VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Martin Šípek

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

AKTIVNÍ DETEKCE TOPOLOGIE LOKÁLNÍ SÍTĚ
DETECTION OF LOCAL AREA NETWORK TOPOLOGY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Martin Šípek**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Zdeněk Martinásek, Ph.D.**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Martin Šípek ***ID:*** 193937
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Aktivní detekce topologie lokální sítě

POKYNY PRO VYPRACOVÁNÍ:

Hlavním cílem práce je návrh a implementace detekčních metod kybernetických útoků mužem uprostřed (MitM) v lokálních sítích LAN (Local Area Network). V teoretické části práce se seznamte s metodami pro detekci a mitigaci těchto útoků a analyzujte dostupné nástroje pro aktivní detekci síťové topologie. V praktické části vytvořte experimentální pracoviště obsahující domácí směrovač Mikrotik, Raspberry Pi (detektor) a osobní počítače představující legitimního uživatele a útočníka. Funkčnost analyzovaných algoritmů pro sledování síťové topologie ověřte a porovnejte (př. výhody nevýhody, rychlost, hardwarové nároky atd.). Navrhněte a implementujte komplexní metodu detekce kybernetických útoků MitM, funkčnost implementace ověřte na experimentálním pracovišti. Detektor bude využívat vlastní implementaci signatury pro systém Suricata, funkční algoritmus detekce síťové topologie nebo vlastní implementaci metody detekce v programovacím jazyku Python.

DOPORUČENÁ LITERATURA:

[1] PFLEEGER, Charles P.; PFLEEGER, Shari Lawrence. Analyzing computer security: a threat/vulnerability/countermeasure approach. Prentice Hall Professional, 2012.

[2] GARCIA-TEODORO, Pedro, et al. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 2009, 28.1-2: 18-28.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Bakalářská práce se zaměřuje na detekci a mitigaci Man-in-the-Middle útoků v lokální síti pomocí vlastní implementace v programovacím jazyku Python. Mezi nejčastější Man-in-the-Middle útoky patří ARP spoofing, který by měl detekční systém identifikovat a následně mitigovat. V teoretické části práce je analyzován současný stav problematiky včetně detailního popsání analýzy sítě a nástrojů k tomu využívaných. Dále jsou popsány kybernetické útoky a to konkrétně útoky Man-in-the-Middle a Denial-of-Service. Praktická část popisuje realizování experimentálního pracoviště a jeho detailně popsané komponenty a instalaci a konfiguraci databáze MySQL a Elasticsearch. Také se zaměřuje na program Suricata, určený k analýze síťového provozu, na vlastní implementaci detekce útoku Man-in-the-Middle a na dosažené výsledky testování realizovaného detekčního systému.

KLÍČOVÁ SLOVA

LAN, IDS, IPS, MitM, DoS, Raspberry Pi, Mikrotik, MySQL, Elastic stack, Suricata, Python

ABSTRACT

The bachelor thesis focuses on the detection and mitigation of Man-in-the-Middle attacks in the local network using its own implementation in the Python programming language. One of the most common Man-in-the-Middle attacks is ARP spoofing, which should be identified by the detection system and then mitigate it. The theoretical part of the thesis analyzes the current state of the issue, including a detailed description of the network analysis and tools which are used in this analysis. Cyber attacks are also described, namely Man-in-the-Middle and Denial-of-Service attacks. The practical part describes the realization of the experimental workplace and its detailed components and the installation and configuration of MySQL and Elasticsearch databases. It also focuses on the Suricata program, designed to analyze network traffic, on the actual implementation of Man-in-the-Middle attack detection and on the achieved results of testing of the implemented detection system.

KEYWORDS

LAN, IDS, IPS, MitM, DoS, Raspberry Pi, Mikrotik, MySQL, Elastic stack, Suricata, Python

ŠÍPEK, Martin. *Detekce aktivní topologie lokální sítě*. Brno, 2020, 54 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Detekce aktivní topologie lokální sítě“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Zdeňkovi Martináskovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod **10**

1 Teoretická část bakalářské práce práce **11**
1.1 ISO/OSI . . . 11
1.2 Metody detekce kybernetických útoků . . . 12
1.2.1 Signature-based a Anomaly-based systémy . . . 13
1.2.2 Host-based a Network-based systémy . . . 13
1.2.3 Packet-based a Flow-based analýza . . . 14
1.2.4 Logování . . . 15
1.3 Metody detekce topologie . . . 16
1.3.1 Detekce topologie na síťové vrstvě . . . 17
1.3.2 Detekce topologie na spojové vrstvě . . . 17
1.3.3 Nástroje určené k detekci topologie . . . 17
1.4 Kybernetické útoky . . . 18
1.4.1 Man-in-the-Middle útoky . . . 18
1.4.2 Denial-of-Service útoky . . . 21

2 Praktická část bakalářské práce **25**
2.1 Experimentální pracoviště . . . 25
2.1.1 Směrovač – MikroTik . . . 25
2.1.2 Detektor – Raspberry Pi . . . 27
2.1.3 Útočník – Kali Linux 2019.4 . . . 30
2.1.4 PC – Windows 10 Pro . . . 30
2.1.5 Databáze logů – MySQL . . . 31
2.1.6 Databáze logů – Elastic stack . . . 33
2.2 Detekce DoS útoku ICMP flood . . . 38
2.3 Detekce MitM útoku ARP spoofing . . . 40
2.4 Arpwatch . . . 44

Závěr **46**

Literatura **48**

Seznam symbolů, veličin a zkratek **52**

Seznam příloh **53**

A Obsah přiloženého CD **54**

Seznam obrázků

1.1 Grafické znázornění modelu ISO/OSI 12
1.2 Packet-based síťový systém 14
1.3 Flow-based síťový systém 15
1.4 Schéma útoku Man-in-the-Middle 19
1.5 Schéma útoku DNS spoofing 20
1.6 Schéma útoku SSL stripping 22
2.1 Blokové schéma experimentálního pracoviště 25
2.2 Reálná podoba experimentálního pracoviště 26
2.3 Nastavení port mirroring na směrovači MikroTik 27
2.4 Výsledné nastavení nftables 30
2.5 Upozornění vygenerované Suricatou 30
2.6 Popis tabulky notifications 32
2.7 Úvodní stránka kibany 36
2.8 Upozornění vygenerované IDS Suricatou 39
2.9 Upozornění vygenerované IPS Suricatou 39
2.10 Upozornění vygenerované Python skriptem během dvoustranného ARP spoofing útoku 44
2.11 Upozornění vygenerované Python skriptem během jednostranného ARP spoofing útoku 44
2.12 Porovnání procentuální spotřeby CPU pythonovského skriptu a arpwatch 45

Seznam tabulek

1.1 Popis paketů použitých v TCP three-way handshake 23
2.1 Technické parametry MikroTik hAP ac^2 26
2.2 Technické parametry Raspberry Pi 4 28

Seznam výpisů

2.1 Konfigurace domácí sítě pro Suricatu 28
2.2 Konfigurace signatur pro Suricatu 29
2.3 Konfigurace programu elasticsearch 34
2.4 Odpověď programu elasticsearch 35
2.5 Konfigurace programu kibana 36
2.6 Konfigurace programu filebeat 38
2.7 Signatura pro detekci útoku ICMP flood 38
2.8 Inicializace funkce sniff 41
2.9 Mikrotik skript k zjištění duplicitních záznamů v ARP tabulce 42
2.10 Blokace komunikace nežádoucí MAC adresy na směrovači Mikrotik . 42

Úvod

Postupem času nezadržitelně vzrůstá popularita připojení k internetu a vzájemné propojování elektroniky přes internet nebo na lokálních sítích. Současně vzrůstá i množství těchto zařízení v domácnostech, a tím pádem vzrůstá i výskyt kybernetických útoků mířených na tato zařízení. Je prakticky nemožné chránit každé koncové zařízení zvlášť, ať už kvůli neustálým aktualizacím na každém přístroji jednotlivě nebo neexistujícím řešením konkrétního typu ochrany před danými útoky. Proto je snaha posunout ochranu z fyzických zařízení na samotnou síťovou úroveň, aby bylo možné zabezpečit všechna zařízení nacházející se v dané síti najednou.

Existují různé typy sítí. Nejmenším typem je LAN síť (Local Area Network), což je kolekce zařízení propojených mezi sebou v určité limitované lokaci. Spojením těchto sítí poté vznikají větší sítě jako například MAN (Metropolitan Area Network) nebo WAN (Wide Area Network). Zabezpečení před kybernetickými útoky na síťové úrovni musí být vždy nastaveno nad typem LAN, jelikož se jedná o privátní síť. Všechny ostatní sítě jsou veřejné [1].

Kybernetických útoků existuje celá řada. Patří mezi ně i útok muž uprostřed, anglicky Man-in-the-Middle (MitM), také známý pod pojmem bucket brigade attack nebo Janus attack. Útočník se v takovém případě, jak název napovídá, nachází mezi dvěma stranami, které věří, že spolu komunikují skrz soukromé spojení, ale ve skutečnosti je celá konverzace kontrolována právě útočníkem. Takový útok může být úspěšný pouze tehdy, pokud útočník vytvoří společnou autentizaci mezi dvěma stranami [2].

Cílem bakalářské práce je návrh a implementace detekční metody kybernetického útoku mužem uprostřed v lokální síti LAN. Práce bude rozdělena na 2 hlavní části. V první části bude popsána teorie, bude analyzován současný stav problematiky a také dostupné nástroje pro aktivní detekci síťové topologie. Druhá část se bude věnovat realizaci experimentálního pracoviště a návrhu detekce útoku mužem uprostřed.

1 Teoretická část bakalářské práce práce

1.1 ISO/OSI

Na začátku vývoje sítí byly jednotlivé LAN sítě nekompatibilní, tudíž se nedaly mezi sebou propojit. Internet tak jak ho známe dnes nemohl tím pádem existovat. Z tohoto důvodu mezinárodní organizace pro normalizaci (International Organization for Standardization), dále jen „ISO“, zavedla otevřený model pod názvem Open System Interconnection Reference Model, neboli „OSI“, který dělí práci v sítích na sedm vrstev [3]. Jeho grafické znázornění lze vidět na obrázku 1.1.

První nebo nejnižší vrstva je **vrstva fyzická**. Jejím úkolem je zajistit přenos jednotlivých bitů mezi odesílatelem a příjemcem po fyzické cestě. Řeší otázky typu, jakým napětím je reprezentována logická jednička a logická nula, jak dlouhé trvání má jeden bit nebo kolik kontaktů mají mít konektory kabelů přenášející dané bity [4].

Druhá vrstva je **vrstva spojová**, někdy také označována jako linková. Má za úkol měnit prostý tok bitů na bezchybný přenos bloků dat, takzvaných rámců. Dále řídí tok dat, detekuje a opravuje chyby rámců, jelikož na fyzické cestě může dojít k rušení či poruchám, a zahajuje, udržuje nebo uzavírá vytvořená spojení. Tato spojení jsou vždy mezi přímo sousedícími komunikačními systémy [4].

Jelikož spojová vrstva poskytuje spojení pouze mezi přímo sousedícími uzly, což mnohdy nemusí stačit, existuje **vrstva síťová**, jejímž úkolem je směrování takzvaných paketů mezi uzly, které mají jeden nebo více mezilehlých uzlů. Zajišťuje tedy volbu nejvýhodnější trasy přes mezilehlé uzly a postupné předávání paketů na těchto uzlech od odesílatele až k příjemci. Dále slouží k síťovému adresování nebo k identifikaci koncových bodů navázaného spojení [4].

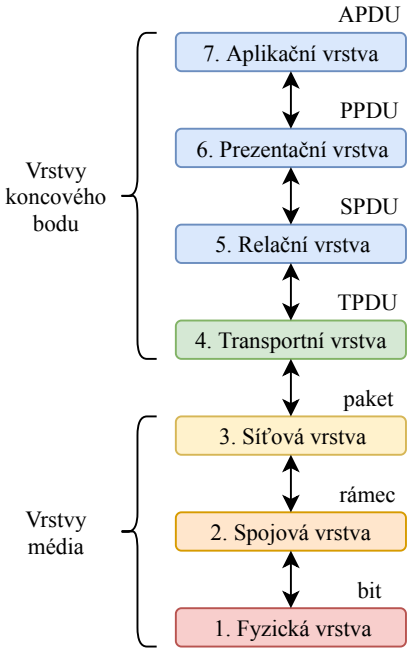
Protože se síťová vrstva stará o přenos paketů mezi jakýmikoliv uzly v síti, vytváří **transportní vrstvě** iluzi, že každý uzel má přímé spojení s libovolným jiným uzlem, tudíž se transportní vrstva stará pouze o komunikaci mezi koncovými body (end-to-end komunikace). Poskytuje kvalitu služby, kterou vyžadují vyšší vrstvy modelu ISO/OSI, to je služba se spojením nebo bez spojení. Protože může být navázáno více spojení, odlišuje jednotlivá spojení pomocí identifikátorů neboli portů [4].

Hlavním úkolem **relační vrstvy** je především synchronizace dat, jejich řízení mezi aplikačními procesy a navazování, udržování a rušení relací mezi koncovými body. V případě poloduplexního spojení určuje, jaká strana má kdy vysílat [4].

Úkolem **prezentační vrstvy** je příprava služeb pro aplikační vrstvu k následné interpretaci přenášených dat. V případě, kdy syntaxe (prezentace) dat na vysílači je jiná než syntaxe dat na přijímači, prezentační vrstva transformuje data z vysílače do přenosové syntaxe, což je dojednaná prezentace dat používaná na přenosové trase,

a na přijímači je transformuje do syntaxe používanou přijímací entitou. Mimo jiné umí tato vrstva data také šifrovat nebo komprimovat [4].

Sedmá a zároveň nejvyšší vrstva je **vrstva aplikační**, která zpřístupňuje informačním systémům OSI. U mnoha systémů se dá uvést, že aplikační vrstva je pouze rozšíření daného operačního systému k využívání síťového prostředí. Funkčně je to velice rozsáhlá vrstva, poskytuje například služby, jako je přenos zpráv nebo identifikace komunikujících uživatelů [4].



Obr. 1.1: Grafické znázornění modelu ISO/OSI

1.2 Metody detekce kybernetických útoků

Analýza sítě je proces získávání informací o probíhající komunikaci na síti z důvodu zefektivnění nebo zvýšení bezpečnosti dané sítě. Především ve velkých firemních sítích je analýza důležitá kvůli přítomnosti velkého množství cenných a citlivých informací. K této analýze slouží nástroj nazývaný Intrusion Detection System (systém detekce vniknutí), neboli „IDS“. Je to systém, který monitoruje a identifikuje počítačové nebo síťové události k určení jakékoliv neobvyklé situace, jenž je posléze považována za neoprávněné vniknutí. Po této definici může vyvstat otázka, jak se tedy liší tento systém od běžného firewallu. Firewall je hardware nebo softwarový program s funkcí zamezit určité komunikaci zakázané předdefinovanou bezpečnostní politikou. Liší se v tom smyslu, že nemá schopnost vyhledávat anomálie do stejné míry jako IDS. Z tohoto důvodu musí být IDS na první linii obrany a pracovat

na stejné úrovni jako firewall. Na rozdíl od firewallů jsou automatizované, jelikož nezávisí na lidských rozhodnutích. Tento systém je pasivní na rozdíl od systému Intrusion Prevention System [5, 6].

Intrusion Prevention System (systém prevence vniknutí), neboli „IPS“ detekuje anomálie podobně jako IDS, ale je navržen tak, aby předcházel vniknutí v reálném čase. Na rozdíl od IDS, který vyvolá pouze poplach v případě vniknutí, IPS blokuje vniknutí nezávisle na lidské interakci. Hlavní nevýhodou tohoto systému jsou vážné důsledky při blokování užitečných přenosů. Oba tyto systémy se mohou rozlišovat podle způsobu detekce nebo místa nasazení [6].

1.2.1 Signature-based a Anomaly-based systémy

Podle způsobu detekce vniknutí mohou být systémy rozděleni na dvě kategorie, signature-based a anomaly-based. Signature-based systémy využívají databázi signatur (vzorů), které odpovídají známé hrozbě, a pokud je nalezena shoda s aktuálním vstupem, je aktivováno varování. Nevýhodou tohoto typu systémů je neschopnost odhalení hrozeb, které nejsou uloženy v databázi, a nutnost konstantní aktualizace těchto databází. Příklady tohoto typu jsou programy Suricata nebo Snort [5, 6].

Anomaly-based systémy si během tréninkové fáze vytvoří model normálního vzoru provozních dat a poté porovnávají nové vstupy podle modelu. Významná odchylka je označena jako anomálie. Přestože signature-based systémy neumí detekovat neznámé hrozby, protože databáze může být zastaralá, mají vysokou přesnost (málo falešných poplachů). Naopak anomaly-based systémy dokáží neznámé hrozby identifikovat, trpí však velikým počtem falešných poplachů. Možný důvod pro takové poplachy může být změna topologie sítě (změna směrování nebo nově připojení hostitelé) [6].

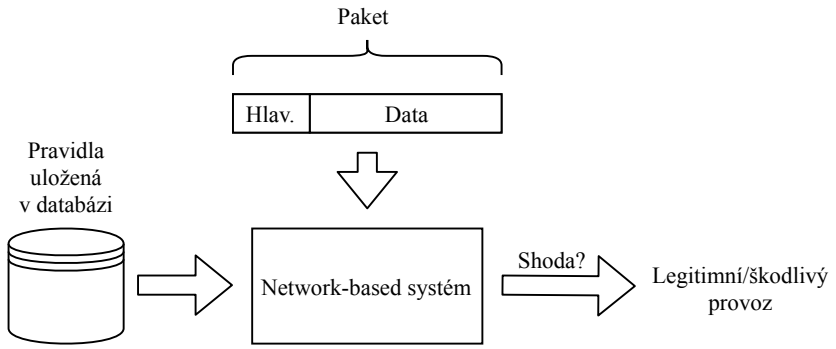
1.2.2 Host-based a Network-based systémy

Dalším způsobem rozdělení těchto systémů je na základě jejich pozice nasazení. Může existovat systém nasazený na koncovém stroji, tedy host-based, který monitoruje jedno zařízení a audituje data, jako například využití zdrojů nebo systémové protokoly, sledované hostitelským operačním systémem [6].

Dalším typem je network-based systém, který monitoruje síť a analyzuje provoz, který protéká určitým segmentem. Na rozdíl od typu host-based, připojení nového hostitele v síti nevyžaduje žádné úsilí navíc, protože je monitorována celá síť, nikoliv jednotlivé koncové stroje. Obecně je snazší aktualizovat jednu komponentu network-based systému než mnoho komponent host-based systému. Existují dvě metody založené na zdroji dat, která mají být analyzována v network-based systému, nazývány packet-based a flow-based [6].

1.2.3 Packet-based a Flow-based analýza

Packet-based analýza je založena na skenování hlaviček jednotlivých paketů, takže se systém rozhoduje o blokaci paketů pouze na základě zdrojové a cílové adresy. Pro použití důkladnější analýzy, kde je potřeba znát mimo hlavičky paketu také jeho obsah, neboli přenášená data, lze využít Deep Packet Inspection („DPI“). V takovém případě je využívána kombinace skenování hlavičky a zároveň i dat pro určení, zda paket blokovat či nikoliv. Příchozí pakety jsou oskenovány a porovnány s každým pravidlem v databázi, jak je ukázáno v obrázku 1.2. DPI pracuje nad sedmou vrstvou modelu ISO/OSI, takže dokáže zjistit podrobné informace o síťovém provozu. Zachycování a analýza paketů může probíhat na různých místech, jako jsou směrovače, přepínače nebo síťové detektory. Hlavní výhoda packet-based analýzy spočívá v tom, že všechny běžné druhy známých útoků lze teoreticky detekovat. Nevýhody této analýzy jsou vyšší náročnost výpočetní techniky, z toho vyplývá zpomalení provozu sítě a neschopnost detekovat neznámé útoky, protože se porovnává s předdefinovanými a známými signaturami. Tato metoda je především využívána signature-based systémy [5, 6].



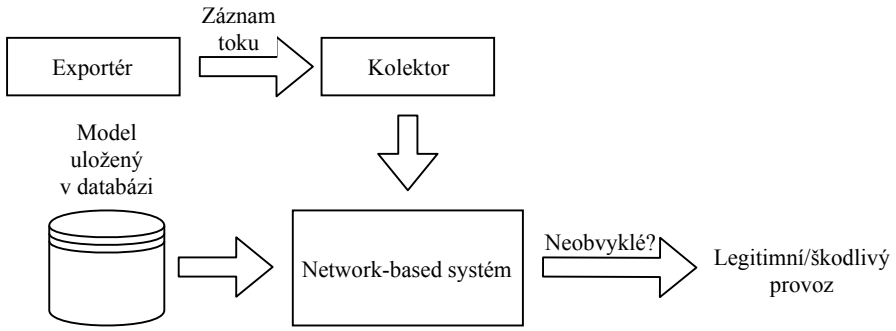
Obr. 1.2: Packet-based síťový systém

Druhá metoda analýzy dat se nazývá Flow-based, jenž je založena na proudech dat. Důležitým faktem o síťovém toku je to, že neposkytuje žádné informace o datech v paketech na rozdíl od packet-based přístupu. Tok lze definovat jako jednosměrný datový proud mezi dvěma komunikačními systémy, kde všechny vysílané pakety tohoto proudu mají následující společné vlastnosti:

- Zdrojová IP adresa
- Cílová IP adresa
- Zdrojový port
- Cílový port
- Protokol

Všechny pakety, které mají těchto pět vlastností totožné spadají do jednoho toku

a v moment, kdy se libovolná vlastnost změní, začne se jednat o nový tok. V současnosti jsou měřicí systémy schopny poskytnout kromě již zmíněných i další charakteristiky jako například počet paketů a množství bajtů přenesených v jednom toku, počáteční čas toku nebo dobu trvání toku. Existuje řada technologií, které vytvářejí data k analýze sítě z datových proudů. Nejpoužívanější a standardní technologií je NetFlow, jenž byla vyvinuta společností Cisco. Lze zabudovat do libovolného směrovače nebo přepínače podporujícího NetFlow a následné nastavení se skládá ze dvou komponent. Je potřeba exportér, který extrahuje hlavičky z každého příchozího paketu a zároveň je zodpovědný za vytvoření záznamu toku a následného odeslání do kolektoru. Kolektor zpracovává a ukládá tyto záznamy přijaté od exportéru, která se následně mohou analyzovat, jak je znázorněno na obrázku 1.3. Jelikož toky poskytují pouze informace o chování připojení, nikoliv samotná data obsažená v paketech, nelze detekovat útoky, které jsou detekovatelné výhradně packet-based analýzou. Jinými slovy, útok, který je obsažen v datech paketu a ne v hlavičce, nebude detekován. Například webové útoky, které jsou založeny na injekci škodlivého kódu na web, mohou být pro flow-based analýzu nedetekovatelnými, jelikož je škodlivý kód viditelný pouze v samotných datech paketu. Takový útok by šel detekovat flow-based analýzou, pokud by útočník provedl více paralelních útoků najednou a způsobil by mnoho toků na server. Výhoda této analýzy je menší velikost zpracovávaných dat na rozdíl od packet-based analýzy a schopnost detekovat doposud neznámé útoky. Hlavní nevýhoda spočívá ve zvýšeném množství falešných poplachů. Tato metoda je především využívána anomaly-based systémy [5, 6].



Obr. 1.3: Flow-based síťový systém

1.2.4 Logování

Výstupem všech výše zmíněných systémů v kapitolách 1.2.1 až 1.2.3 jsou takzvané logovací soubory, což jsou soubory záznamů popisující konkrétní události, které nastaly ve sledovaném systému. Takové záznamy lze rozdělit do následujících kategorií:

- **Informační** – popisují stav nebo událost

- **Ladicí** – používají se při vývoji
- **Varovné** – upozorňují na problém, který není natolik závažný, aby se systém zhroutil
- **Chybové** – upozorňují na problém, který může ohrozit funkčnost systému
- **Pohotovostní** – popisují událost spojenou s bezpečností systému

Obecně každý informační systém produkuje logovací soubory a přestože neexistují žádné standardy na jejich tvorbu, což způsobuje obrovské potíže s jejich analyzováním, skoro každý logovací soubor obsahuje časové razítko, které nese informace o době, kdy k události došlo, zdroj vyvolávající záznam události a konkrétní popis události, kde lze najít podrobné informace o stavu problému. Všechny tyto informace se ukládají buď v textovém nebo binárním formátu a oba přístupy mají své výhody i nevýhody. Textový formát logovacích souborů je snadno čitelný v základních textových programech, vytvoření nespotřebovává tolik výpočetního výkonu jako u binárního formátu a má společnou syntaxi pro veliké množství aplikací. Binární formát je lehce komprimovatelný a jednodušeji se dá uložit do databází [7, 8, 9].

1.3 Metody detekce topologie

Topologie sítě je uspořádání různých síťových prvků. Její detekce je nesmírně důležitá, jelikož poskytuje informace o tom, zda-li síť funguje správně nebo jestli nějaké konkrétní zařízení funguje tak, jak má. Topologie sítě může mít velký vliv na to, jak síť funguje, co se stane, když zařízení přestane fungovat, nebo jak složité je síť spravovat. Topologie je obvykle reprezentována grafem, ve kterém jsou síťová zařízení představována jako uzly a spojení mezi těmito zařízeními jsou představována jako linie mezi uzly. Existují dva různé typy topologií sítí:

- **Fyzická topologie sítě** – umístění různých komponentů sítě.
- **Logická topologie sítě** – znázornění jak proudí data v síti.

Hlavní fyzické topologie sítí jsou:

- **Sběrnice** – síť je nastavena v přímé linii, která umožňuje datům protékat sítí z jednoho zařízení do každého uzlu postupně.
- **Kruh** – síť je uspořádána do kruhu, takže data mohou kolem kruhu protékat jedním nebo oběma směry
- **Strom** – síť je ve tvaru pomyslného stromu, tudíž má jeden kořenový uzel, ze kterého se rozrůstají větve s dalšími uzly. Pokud přestane fungovat uzel v jedné větvi, ostatní větve jsou tímto výpadkem nedotčeny na rozdíl od topologií sběrnice a kruhu, kde by výpadek jednoho uzlu mohl způsobit pád celé sítě.
- **Hvězda** – síť má jeden centrální uzel, ke kterému jsou připojeny všechny ostatní uzly.

- **Smíšená** – síť, kde je každý uzel propojen se všemi ostatními uzly, což umožňuje přesměrovat data k určenému cíli při selhání jednoho z uzlů.
- **Hybridní** – kombinace výše zmíněných topologií [10, 11].

1.3.1 Detekce topologie na síťové vrstvě

Na síťové vrstvě existuje standardní protokol, který je velice užitečný pro detekci topologie. Jedná se o ICMP (Internet Control Message Protocol), jenž poskytuje zajímavé informace o sítích třetí vrstvy. Jedním z běžných nástrojů používaných k identifikaci síťových skoků je `traceroute`. Díky němu lze najít cesty, kterými proudí pakety, a objevit směrovače. Dalším nástrojem, který se může využít při detekci topologie síťové vrstvy je `ping`. Při odeslání požadavku na broadcastovou adresu nástroj `ping` obdrží odpověď od všech zařízení v síti [10].

1.3.2 Detekce topologie na spojové vrstvě

Pro detekci topologie na spojové vrstvě existuje několik protokolů, z nichž dva nejznámější jsou:

- **LLDP (Link Layer Discovery Protocol)** – protokol používán síťovými zařízeními k propagaci jejich identity, schopností a sousedů v lokální síti, což umožňuje objevovat sousední uzly.
- **CDP (Cisco Discovery Protocol)** – protokol vyvinutý společností Cisco, jenž je podporován i jinými výrobci síťových zařízení.

Tyto protokoly lze použít k identifikaci toho, kdo je připojen k určitému síťovému rozhraní poslouchánímdat LLDP nebo CDP. Většina přepínačů podporuje jeden nebo oba dva protokoly. Dokonce i operační systémy založené na Linuxu a operační systém Windows podporují protokol LLDP. Na Windows avšak není vestavěný nástroj, který by poskytoval tyto informace, tudíž je nutné nějaký doinstalovat [10].

Dalším složitějším způsobem detekce topologie na spojové vrstvě je analyzování ARP (Address Resolution Protocol) tabulek každého směrovače, aby se zjistilo, kde jsou připojená další zařízení. Tento způsob je ovšem brán v potaz jako poslední možnost, protože stojí veliké úsilí [10].

1.3.3 Nástroje určené k detekci topologie

Detekce topologie sítě je důležitá pro udržení nejvyššího možného výkonu nebo identifikaci úzkých hrdel v síti. Přistupovat k tomuto procesu manuálně může být ovšem velmi časově náročné, pokud se jedná o velkou nebo komplikovanou síť. Proto existuje řada nástrojů, které tento proces usnadňují [11].

Network Topology Mapper je program vyvinutý společností SolarWinds, jenž vyniká ve schopnosti automaticky vykreslit topologii sítě. Funkce automatické detekce je obzvlášť užitečná, pokud již existuje vytvořená síť, kterou je potřeba správně detekovat a vykreslit do grafu. Automatická detekce také zajišťuje aktualizaci vykreslené topologie v případě jakékoliv změny. Program lze nakonfigurovat, podle jakého výše zmíněného protokolu bude detekovat topologii [11].

Device42 je databáze s nástroji pro automatické detekce topologie. Dokáže identifikovat zařízení s přiřazenou IP adresou i bez ní, sledovat vzájemné závislosti mezi zařízeními nebo jejich využití zdrojů. Jedná se o užitečný nástroj, ale automatickou detekci může být obtížné implementovat, pokud je síť velmi složitá, protože má tendenci shromažďovat nepotřebné informace [11].

Intermapper umožňuje rychle a efektivně vytvářet vlastní grafy a automaticky detekovat všechna zařízení s přiřazenou IP adresou. Následně lze tyto grafy upravovat podle preferencí uživatele. Poskytuje textová nebo emailová upozornění v reálném čase, pokud se vyskytnou nějaké problémy [11].

Nmap je open-source síťový detekční nástroj, který používá ICMP pakety k určení, jaká zařízení se nacházejí v síti, jaké služby jsou těmito zařízeními nabízeny nebo jaký operační systém na daném zařízení běží. Nabízí obrovské množství možností, proto je časově náročné naučit se používat tento program [11].

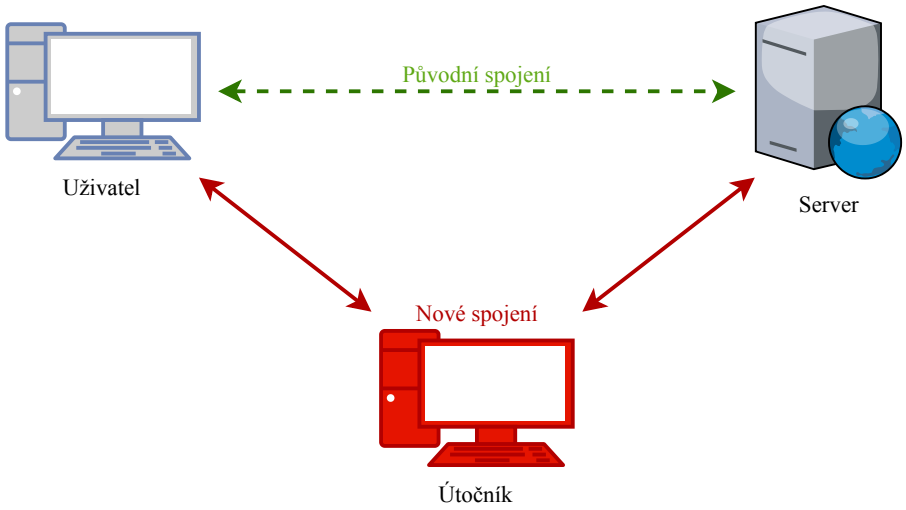
1.4 Kybernetické útoky

Kybernetické útoky jsou škodlivé a záměrně mířené pokusy jednotlivců nebo organizovaných skupin narušit informační systémy jiné osoby nebo organizace a to se záměrem nějaké výhody. Někteří útočníci považují útočení a následné vymazávání systémů nebo databází za formu takzvaného hacktivismu. Jiní útočníci se snaží finančně obohatit požadováním výkupného od jejich oběti. Kyber kriminalita narůstá každý rok a padesát tři procent všech kybernetických útoků mělo za následek škodu v hodnotě pět set tisíc dolarů nebo víc. Běžné typy jsou malware, phishing, Man-in-the-Middle útok, Denial-of-Service útok, SQL injection nebo Zero-day exploit [12, 13].

1.4.1 Man-in-the-Middle útoky

Man-in-the-Middle útoky, dále jen „MitM“, jsou běžným typem útoků na kybernetickou bezpečnost, který útočníci provádějí, aby vložili svou přítomnost mezi komunikaci dvou stran za účelem získání přístupu ke všem zaslaným informacím. To se může stát v jakékoliv formě online komunikace. Útočníci mohou zabránit uživatelům v odesílání a příjímání dat nebo dokonce mohou přesměrovat zprávy jinému

uživateli. Narušená komunikace je zbavena veškerého šifrování, aby ji útočníci mohli odposlechnout, změnit nebo přesměrovat a následně je znovu zašifrována a odeslána k zamýšlenému zdroji. Z tohoto důvodu může být velice obtížné takovýto útok detekovat. Hlavním cílem útoku MitM je odposlechnout konverzaci uživatelů a maskovat útočníkovu přítomnost tak, aby se zdálo, že se komunikace neúčastní žádná třetí osoba. Schéma útoku je zobrazeno na obrázku 1.4. Útok se skládá ze dvou fází. První fáze je takzvané zachycení, což je fáze, kdy je uživatelův provoz zastaven sítí útočníka. Zachycení může být provedeno jedním z následujících útoků [14, 15, 16, 17].



Obr. 1.4: Schéma útoku Man-in-the-Middle

ARP spoofing

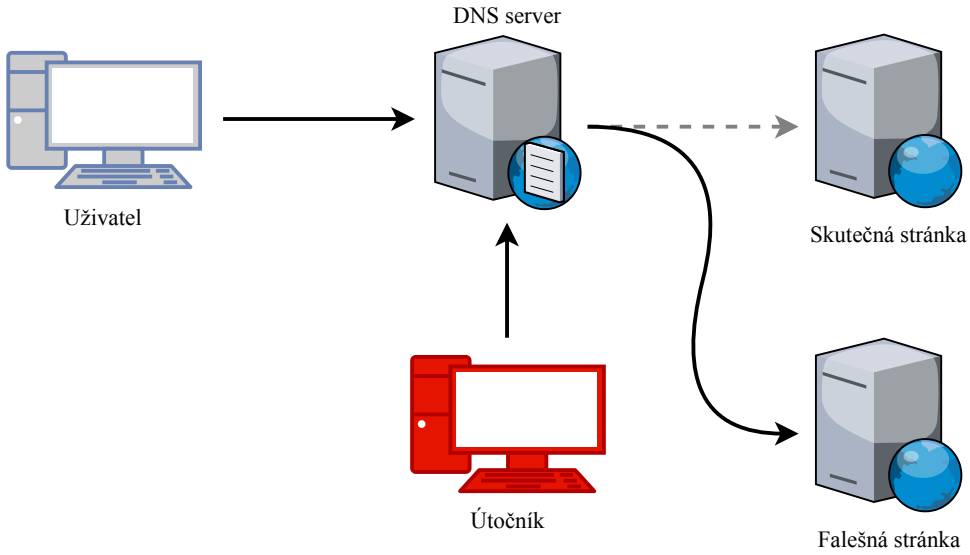
ARP spoofing, také nazýván ARP poisoning, využívá protokol ARP (Address Resolution Protocol), což je protokol, který získává fyzickou MAC adresu na základě IP adresy. Útočník tohoto využije tím, že propojí svou MAC adresu s legitimní IP adresou v lokální síti pomocí falešných ARP zpráv, takže může přijímat data určená pro vlastníka přidruženého k této IP adrese [14, 17, 18].

IP spoofing

IP spoofing je jednou z nejčastějších forem online kamufláže. Jedná se o metodu, při které útočník maskuje svou identitu tím, že vytváří pakety s upravenou zdrojovou adresou legitimního zařízení. Tímto způsobem obě komunikující strany pošlou své pakety útočníkovi místo toho, aby je přímo poslali na skutečné místo určení [19, 20, 21].

DNS spoofing

DNS spoofing, také nazývaný DNS cache poisoning, je typ útoku, který využívá systémové zranitelnosti na DNS serveru k odklonění provozu od legitimních serverů a nasměrování ho k falešným. DNS, neboli Domain Name System, je protokol, který převádí názvy domén na IP adresy odpovídajícímu serveru. Je to jeden z nejdůležitějších infrastrukturních protokolů na internetu s cílem usnadnit komunikaci a zbavit lidi problému se zapamatováním IP adres každého serveru, ke kterému by chtěli přistupovat. Po zadání adresy domény do prohlížeče se odešle žádost o překlad názvu na server DNS, který poté vyhledá název domény ve svém adresáři a vrátí IP adresu odpovídajícího serveru. Při DNS spoofingu útočník zachytí požadavek na DNS server a vrátí oběti falešnou IP adresu. Schéma tohoto útoku je zobrazeno na obrázku 1.5 [22, 23, 24].



Obr. 1.5: Schéma útoku DNS spoofing

Jakmile je odposlech úspěšně proveden nastává druhá fáze útoku MitM, dešifrování, které je potřeba provést bez jakéhokoliv upozornění oběti, což se dá učinit následujícími způsoby.

HTTPS spoofing

V případě, kdy oběť přistupuje k webovému serveru prostřednictvím HTTP (nezabezpečeného) protokolu a server podporuje HTTPS (zabezpečený) protokol, je automaticky přesměrován na tento zabezpečený web. Při útoku HTTPS spoofing je právě v tento moment odeslán falešný certifikát do prohlížeče oběti místo legitimního certifikátu webového serveru. Tento falešný certifikát je v souladu s daným serverem, tudíž předčí ověření se standardy důvěryhodných webů provedené prohlížečem.

Tím pádem útočník komunikuje s obětí díky svému falešnému certifikátu, takže je schopný dešifrovat zprávy posílané obětí na svém zařízení, přečíst si je nebo upravit a následně je přeposílat zašifrovanou komunikací na webový server [14, 17, 25].

SSL hijacking

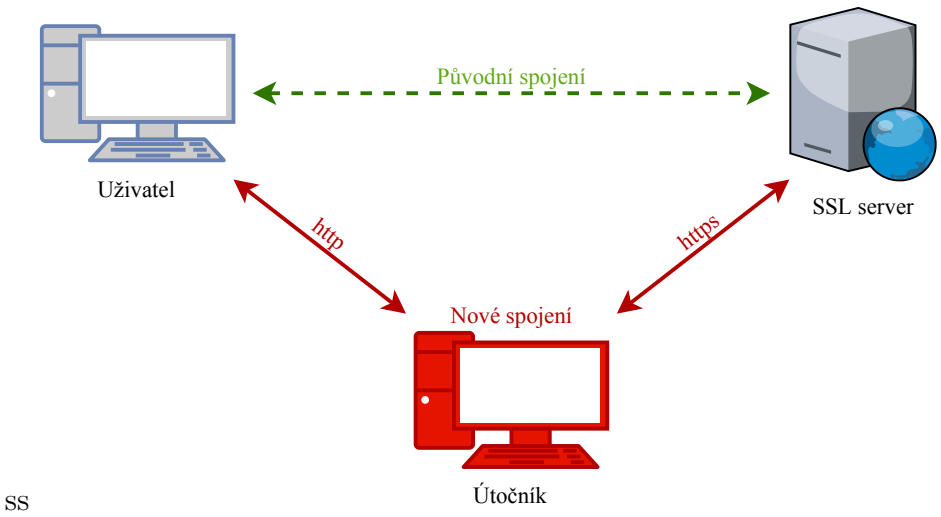
Při útoku SSL (Secure Sockets Layer) hijacking útočník předá falešné autentizační klíče oběti i serveru během procesu TCP handshake, aby převzal kontrolu nad celou relací, zatímco uživatel předpokládá, že jde o zabezpečené spojení [14, 17].

SSL stripping

Útok SSL stripping nevyužívá žádné zranitelnosti SSL protokolu, ale je navržen tak, aby navedl naivního uživatele k použití nezabezpečeného protokolu HTTP namísto zabezpečeného HTTPS. Protokol SSL vytváří zabezpečený kanál mezi uživatelem a serverem, ale v tomto případě se tento kanál vytvoří pouze mezi útočníkem a serverem a mezi obětí a útočníkem zůstane nezabezpečený kanál bez SSL protokolu. Aby útočník odstranil SSL, musí zasáhnout do přesměrování HTTP na zabezpečený protokol HTTPS a zachytit požadavek uživatele na server. Útočník bude nadále navazovat zabezpečené spojení mezi sebou a serverem a nezabezpečené spojení s uživatelem. Schéma tohoto útoku je zobrazeno na obrázku 1.6 Nejjednodušší je ukázat si problematiku na příkladu. Alice jakožto oběť chce poslat peníze skrz bankovní zabezpečený server, ale Eva, jenž je útočník, chce tuto komunikaci zachytit a získat přihlašovací jméno a heslo Alice. Za tímto účelem naváže spojení s Alicí, čímž přeruší její komunikaci se serverem a v moment, kdy Alice požádá o přístup k bankovnímu serveru ve svém prohlížeči, neodešle tento požadavek bance, nýbrž Evě, která ho následně přepošle bance. Server reaguje na Evinu, původně Alicinu žádost ve formě adresy HTTPS. Tu Eva degraduje na nezabezpečenou HTTP adresu a přepošle jí Alici. Tím pádem vše, co Alice pošle, bude ve formě čistého textu a Eva si může zobrazit citlivé informace jako například přihlašovací údaje nebo informace o kreditní kartě. Útok je úspěšně dokončen [26, 27].

1.4.2 Denial-of-Service útoky

Útoky typu Denial-of-Service, dále jen „DoS“, jsou útoky, jejichž cílem je vypnout zařízení nebo síť, čímž je pro určené uživatele nedostupná. DoS útoky tohoto dosáhnou zaplavením cíle provozem nebo zasláním informací, které způsobí selhání. V obou případech útok připravuje legitimní uživatele o službu, kterou očekávali. Oběti DoS útoků jsou často webové servery nadnárodních organizací, jako jsou bankovní, obchodní nebo mediální společnosti nebo státní organizace. Ačkoli tyto útoky



Obr. 1.6: Schéma útoku SSL stripping

obvykle nevedou k odcizení nebo ztrátě důležitých informací nebo jiných aktiv, mohou oběti stát mnoho času a peněz, než je vyřeší. Existují dvě obecné metody DoS útoků: záplavové útoky a logické útoky. Konceptem u záplavových útoků je zahltit vyrovnávací paměť systému velikým množstvím dat, což má za následek zpomalení nebo úplné ukončení serveru. Vyrovnávací paměť (buffer) je dočasná oblast pro ukládání dat a pokud systém dostane více dat než bylo původně přiděleno k uložení, přebývající data „přetečou“, což způsobí, že se některá data dostanou do jiných vyrovnávacích pamětí, a tím pádem mohou být poškozena nebo přepsána data, které byla v dané paměti uložena. Typičtí zástupci záplavových útoků jsou například ICMP flood nebo SYN flood [28, 29].

ICMP flood

ICMP flood (záplava) útok, známý také jako smurf attack (šmoulí útok) nebo ping of death (cinknutí smrti), je útok, při kterém se útočník pokouší zahltit cílové zařízení pomocí ICMP paketů, což způsobí, že se cíl stane nedostupný. ICMP (Internet Control Message Protocol) je protokol třetí vrstvy, který síťová zařízení používají ke komunikaci. Diagnostické nástroje jako traceroute nebo ping používají právě ICMP pakety za účelem diagnostiky stavu a konektivity zařízení nebo spojení mezi odesílatelem a příjemcem. Požadavek ICMP vyžaduje výpočetní prostředky serveru, aby zpracovávali každou příchozí zprávu i odchozí odpověď, a to stejné platí i pro šířku pásma. Při dostatečném množství přicházejících ICMP zpráv se server zahltí a stane se nedostupný pro legitimní uživatele [30].

SYN flood

SYN flood útok využívá zranitelnosti v procesu navazování TCP spojení, který je nazýván three-way handshake (třícestné podání rukou). Jedná se o tříkrokový proces, který vyžaduje, aby si uživatel a server vyměnili synchronizační (SYN) a potvrzovací (ACK) pakety před zahájením procesu skutečné datové komunikace. V tabulce 1.1 jsou popsány všechny typy paketů, které jsou použity v průběhu procesu three-way handshake [31].

Jméno	Popis
SYNchronize	Slouží k zahájení a navázání spojení.
ACKnowledgment	Slouží k potvrzení přijetí SYN paketu.
SYN-ACK	SYN odpovídající strany a ACK předchozího paketu

Tab. 1.1: Popis paketů použitých v TCP three-way handshake

V prvním kroku klient naváže spojení se serverem odesláním SYN paketu, který obsahuje náhodně vygenerované sekvenční číslo x a potvrzovací číslo 0. V druhém kroku server odpovídá na požadavek paketem SYN-ACK, jenž obsahuje náhodně vygenerované sekvenční číslo serveru y a potvrzovací číslo $x + 1$. Ve třetím kroku klient odpovídá paketem ACK obsahujícím číslo sekvence $x + 1$ a potvrzovací číslo $y + 1$, čímž se vytvoří stabilní spojení a zahájí se proces skutečného přenosu dat [31, 32].

U samotného SYN flood útoku útočník posílá na cílený server velké množství SYN paketů často s podvrženými IP adresami. Server poté odpoví na každou žádost o navázání spojení paketem SYN-ACK a ponechá port otevřený a připravený k přijetí následné odpovědi, která samozřejmě nikdy nepřijde. Zatímco server čeká na konečný ACK paket, útočník pokračuje v odesílání dalších SYN paketů. Příchod každého nového paketu způsobí, že server dočasně udržuje nový port otevřený po určitou dobu a jakmile budou využity všechny dostupné porty, server nebude schopen nadále fungovat [32, 33].

Druhým typem DoS útoků jsou logické útoky, které jednoduše zneužívají zranitelnosti, které způsobují selhání cílového systému nebo služby. Při těchto útocích je odeslán input (vstup) uživatele, který využívá chyby v cíli, jenž se následně zhroutí nebo je vážně poškozen, takže k němu nelze přistupovat. Mezi nejznámější logické útoky patří Slowloris, Land attack nebo XMasTree attack [28, 32].

Distributed-Denial-of-Service

Dalším typem DoS útoku je takzvaný Distributed-Denial-of-Service útok, dále jen „DDoS“. K takovému útoku dochází, když velká skupina systémů uskuteční synchro-

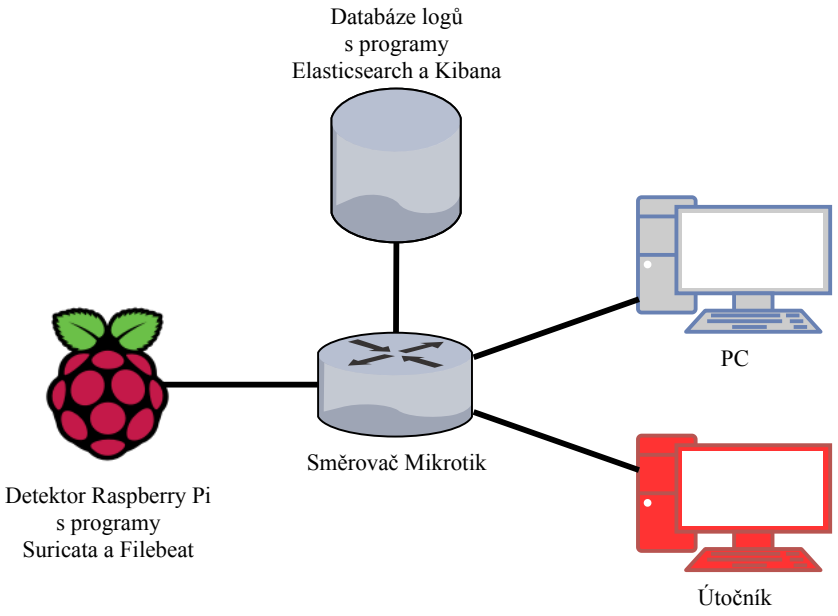
nizovaný útok DoS na jediný cíl. Zásadní rozdíl mezi DoS a DDoS spočívá v tom, že místo útoku z jednoho systému je cíl napaden z mnoha systémů najednou. Taková skupina se nazývá botnet, což je obrovské množství systémů napadených malwarem, tudíž je útočník může ovládat z jednoho master systému. To poskytuje útočníkovi několik výhod. Může využít větší výpočetní výkon, zdroj útoku je obtížné odhalit kvůli náhodnému rozmístění útočných systému nebo je velice náročné identifikovat samotného útočníka, protože se „skrývá“ za obrovským množstvím kompromitovaných systémů. Moderní bezpečnostní technologie vyvinuly mechanismy na obranu proti většině forem DoS útoků, ale vzhledem k jedinečným vlastnostem DDoS útoků jsou stále považovány za zvýšenou hrozbu [28].

2 Praktická část bakalářské práce

Hlavním cílem praktické části je vytvoření experimentálního pracoviště a následný návrh a implementace signatury do programu Suricata nebo skriptu v programovacím jazyku Python na detekci útoku Man-in-the-Middle.

2.1 Experimentální pracoviště

Experimentální pracoviště představuje lokální síť tvořenou domácím směrovačem hAP ac^2 od firmy MikroTik. Na ten je připojen detektor Raspberry Pi, jenž má nainstalovaný software Suricata pro detekci síťového provozu založenou na bázi signatur. Dále jsou na směrovač připojeny dva počítače. Počítač označen jako PC představuje legitimního uživatele s operačním systémem Windows 10 Pro. Útočník používá operační systém Kali Linux ve virtuálním prostředí běžícím v programu VirtualBox. V neposlední řadě je na směrovač připojena databáze MySQL běžící na serveru s operačním systémem Ubuntu 20.04, jenž také funguje ve virtuálním prostředí. Blokové schéma topologie je zobrazeno na obrázku 2.1 a reálná podoba experimentálního pracoviště na obrázku 2.2



Obr. 2.1: Blokové schéma experimentálního pracoviště

2.1.1 Směrovač – MikroTik

MikroTik je Lotyšská společnost založena roku 1996 za účelem vývoje směrovačů. Nyní poskytuje jak software v podobě operačního systému RouterOS, který nabízí

Obr. 2.2: Reálná podoba experimentálního pracoviště

rozsáhlé ovládací prvky, stabilitu a flexibilitu, tak i hardware s názvem RouterBOARD pro připojení k internetu do celého světa. V této práci je použit model hAP ac^2 jehož technické parametry jsou zobrazeny v tabulce 2.1. Disponuje pěti ethernetovými porty, jedním pro připojení k WAN síti a čtyři k LAN, a duální anténou pro 2,4 GHz a 5 GHz pásma k bezdrátovému připojení k internetu. Směrovač má také USB port jenž může sloužit k připojení externího úložiště nebo 4G/LTE modemu. Jednou z mnoha funkcí je `port mirroring`, což je funkce umožňující kopírovat pakety z jednoho rozhraní na rozhraní jiné. V případě tohoto experimentálního pracoviště budou pakety kopírovány z portu, ke kterému je připojen útočník, na port, který náleží detektoru [34]. Nastavení této funkce je popsáno v následujícím odstavci.

Operační systém	RouterOS
Procesor	716 MHz 32 bit Quad core ARM Cortex A7
Paměť	128 MB
Úložiště	16 MB

Tab. 2.1: Technické parametry MikroTik hAP ac^2

Funkci `port mirroring` je nutno nastavit, aby všechny pakety na portu útočníka, v tomto experimentálním pracovišti je to konkrétně port 2, byly zkopírovány na port detektoru, port číslo 5. Tohoto nastavení lze docílit buď v grafickém rozhraní směrovače, v záložce `Switch`. Ve sloupci `Mirror Source` je potřeba zvolit `ether2`

a sloupec `Mirror Target` bude mít zvolenou možnost `ether5`. Finální nastavení je viditelné na obrázku 2.3.

Obr. 2.3: Nastavení port mirroring na směrovači MikroTik

Další možný postup je přes příkazovou řádku směrovače. Po zadání následujícího příkazu lze docílit stejného výsledku.

```
set switch1 mirror-source=ether2 mirror-target=ether5
```

2.1.2 Detektor – Raspberry Pi

Raspberry Pi je název řady jednodeskových počítačů vytvořených britskou charitativní organizací Raspberry Pi, jejímž cílem je vzdělávat lidi v práci na počítači a vytvářet snadnější přístup k počítačovému vzdělání. První verze vydaná v roce 2012 měla 700 MHz jednojádrový procesor a pouze 256 MB RAM paměť. Technické

parametry Raspberry Pi použitého v této práci a zároveň nejnovějšího modelu lze vidět v tabulce 2.2. K ovládání tohoto počítače je potřeba monitor/televize, klávesnice a myš nebo se lze připojit vzdáleně přes lokální síť s použitím programu VNC Viewer, který je v počítači již předinstalován. Aby Raspberry Pi fungovalo jako detektor je třeba nainstalovat IPS Suricata. Návod instalace je popsán v následující sekci.

Operační systém	Raspbian
Procesor	1,5 GHz 64 bit Quad core ARM Cortex A72
Paměť	4 GB
Úložiště	Micro-SD slot

Tab. 2.2: Technické parametry Raspberry Pi 4

Suricata

Suricata je volně dostupný open source nástroj na detekci síťových hrozeb vlastněný neziskovou organizací Open Information Security Foundation (OISF). Může být nakonfigurována buď jako IDS nebo IPS a je kompatibilní s UNIXovými systémy i systémem Windows. Síťový provoz analyzuje na bázi signatur a její hlavní výhodou oproti ostatním IDS/IPS systémům je, že dokáže fungovat na více vláknech, tudíž může kontrolovat více paketů najednou.

Suricatu ve verzi 5.0.0 je možné stáhnout z oficiálních stránek [38] a nainstalovat dle návodu [39]. Po úspěšné instalaci je potřeba systém nakonfigurovat. Všechno konfigurační nastavení je uloženo v souboru `/etc/suricata/suricata.yaml`. Na začátku souboru je část, kde lze definovat domácí síť, jenž je vidět ve výpise 2.1.

Výpis 2.1: Konfigurace domácí sítě pro Suricatu

```
vars:
    # more specific is better for alert accuracy and \
    performance
    address-groups:
        HOME_NET: "[192.168.88.0/24]"
        EXTERNAL_NET: "!$HOME_NET"
```

Z výpisu je patrné, že se domácí síť nachází na adrese 192.168.88.0/24 a vše ostatní spadá pod vnější síť. Dalším bodem, který se musí nastavit, je výchozí umístění signatur, jelikož při instalaci se všechna pravidla stáhla do složky `/var/lib/suricata/rules`, ale v konfiguračním souboru je uvedena složka `/etc/suricata/rules`. Navíc je nutné nastavit cestu k souboru `local.rules`, ve kterém budou

specifikována vlastní pravidla. Všechny provedené změny jsou viditelné ve výpise 2.2.

Výpis 2.2: Konfigurace signatur pro Suricatu

```
default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules
  - local.rules
```

Suricata má nastavenou adresu pro domácí (důvěryhodnou) síť a také cestu k souborům, ve kterých jsou uloženy výchozí i vlastní signatury. Suricatu lze zapnout v IDS módu. Pro převod systému do IPS je třeba nastavit nftables, díky čemuž přesměrujeme síťový provoz do Suricaty. Nftables mají výhodu v tom, že mohou implementovat firewall, tudíž je možnost, aby pakety prošly nejdříve pravidly ve firewallu a až po tom byly poslány k analýze Suricatou. Toto nastavení se provede následujícím způsobem.

```
pi@raspberrypi:~ $ sudo iptables -I INPUT -j NFQUEUE
pi@raspberrypi:~ $ sudo iptables -I OUTPUT -j NFQUEUE
pi@raspberrypi:~ $ sudo nft -i
nft> add table filter
nft> add chain filter IPS { type filter hook forward \
     priority 10;}
nft> add rule filter IPS queue
```

Výsledné nastavení nftables je zobrazeno na obrázku 2.4.

Po provedení všech potřebných změn je Suricata připravena ke spuštění v IPS módu. To se provede příkazem:

```
sudo suricata -c /etc/suricata/suricata.yaml -q 0
```

Pro spuštění Suricaty v IDS módu se používá tento příkaz.

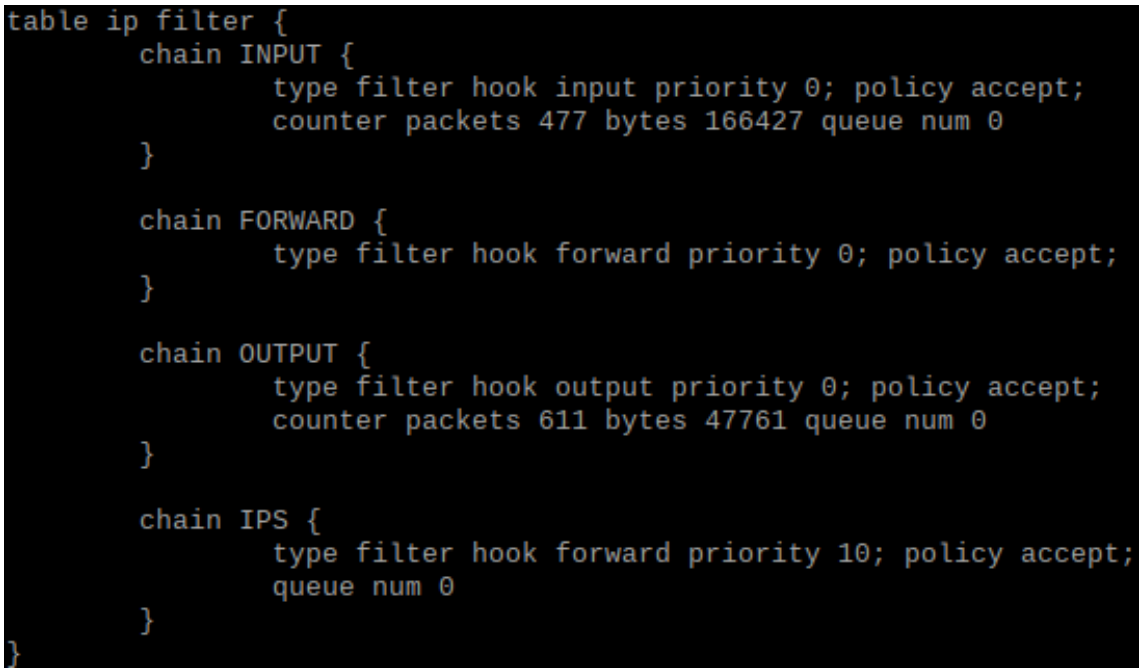
```
sudo suricata -c /etc/suricata/suricata.yaml -i eth0
```

Ke kontrole, zda je vše správně nakonfigurováno a Suricata běží v pořádku, lze využít příkaz

```
curl testmyids.com
```

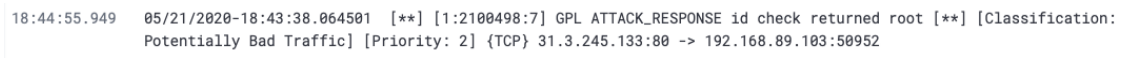
Po jeho zadání by měla Suricata vygenerovat upozornění. Všechna upozornění týkající se kybernetických útoků nebo jiných hrozeb se ukládají do souboru `/var/log/suricata/fast.log`. Ten se může zobrazit například příkazem

```
sudo tail -f /var/log/suricata/fast.log
```



Obr. 2.4: Výsledné nastavení nftables

Vygenerované upozornění lze vidět na obrázku 2.5.



Obr. 2.5: Upozornění vygenerované Suricatou

2.1.3 Útočník – Kali Linux 2019.4

Kali Linux běžící ve virtuálním prostředí aplikace VirtualBox na útočníkově počítači ve verzi 2019.4 je speciální linuxová distribuce založená na distribuci Debian. Využívá se především na penetrační testování nebo bezpečnostní audit. Je vyvíjen společností Offensive Security, jenž poskytuje školení v oblasti informační bezpečnosti. Kali Linux obsahuje více než šest set penetračních nástrojů. Jedním z nich je například `hping3`, nástroj na sestavování nebo analyzování paketů. Název napovídá, že půjde především o ICMP pakety, ale podporuje i TCP nebo UDP. Může být využit například k ICMP flood útoku [36].

2.1.4 PC – Windows 10 Pro

Počítač s operačním systémem Windows 10 Pro představuje v lokální síti legitimního uživatele. Bude využit jako cíl kybernetických útoků použitých k otestování implementace vlastních signatur pro systém Suricata.

2.1.5 Databáze logů – MySQL

Databáze MySQL je nainstalována na serveru Ubuntu 20.04 ve virtuálním prostředí VirtualBox. Slouží k centralizaci logů vygenerovaných detektorem. Instalace a konfigurace této databáze je relativně snadná. Instalaci lze provést příkazem

```
sudo apt install mysql-server
```

V momentě, kdy je instalace úspěšně dokončena, se databáze sama automaticky spustí. Pro kontrolu, zda vše proběhlo v pořádku a databáze běží, lze zadat příkaz

```
sudo systemctl status mysql
```

Pokud je databáze spuštěna, je potřeba spustit skript, který za nás provede základní bezpečnostní opatření jako například nastavení hesla pro uživatele root pro přístup do databáze, smazání databáze test nebo zakázání přístupu k databázi uživatele root ze vzdáleného zařízení. Tento skript lze spustit příkazem

```
sudo mysql_secure_installation
```

Aby bylo možné se připojit k databázi ze vzdáleného zařízení, je potřeba nakonfigurovat několik nastavení. Nejdříve se musí zakomentovat řádek `bind-address` v souboru `/etc/mysql/mysql.conf.d/mysqld.cnf`, jenž povoluje připojení k databázi pouze z lokálního zařízení. Pro aplikaci provedených změn je nezbytné databázi restartovat příkazem

```
sudo systemctl restart mysql.service
```

Dále je nutné se přihlásit a vytvořit uživatele, kterému se udělí práva přístupu k databázi z jakéhokoliv zařízení. Přihlásit se je možné příkazem

```
sudo mysql
```

a vytvoření uživatele se provede následovně

```
CREATE USER 'name'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'name'@'%';
```

, kde `name` označuje jméno uživatele a `password` heslo, pod kterým se uživatel bude přihlašovat do databáze. `%` neidentifikuje žádnou specifickou IP adresu, takže uživatel se může přihlásit odkudkoliv a `*.*` označuje všechny databáze a všechny tabulky.

V momentě, kdy je databáze MySQL úspěšně nainstalována a nakonfigurována, lze vytvořit databázi a tabulku, do které se budou ukládat logy vygenerované detektorem. To se může provést následujícími příkazy

```
CREATE DATABASE arpdetection;
USE arpdetection;
CREATE TABLE notifications(
id INT AUTO_INCREMENT PRIMARY KEY,
type VARCHAR(20),
time TIMESTAMP,
message VARCHAR(255));
```

Těmito příkazy se vytvořila databáze nazvaná `arpdetection` a uvnitř této databáze byla vytvořena tabulka `notifications` se sloupci `id`, což je rostoucí identifikační číslo pro každý záznam v tabulce, `type` je text o maximální délce 20 značek, který označuje druh upozornění, `time` je čas, kdy byl log vygenerovaný a sloupec `message` obsahuje zprávu o maximální délce 255 znaků, která popisuje okolnosti, proč byl tento log vygenerován. Popis tabulky lze vidět na obrázku 2.6.

```
mysql> describe notifications;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int          | NO   | PRI | NULL    | auto_increment |
| type    | varchar(20)  | YES  |     | NULL    |                |
| time    | timestamp    | YES  |     | NULL    |                |
| message | varchar(255) | YES  |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Obr. 2.6: Popis tabulky notifications

Nyní je MySQL databáze připravena k používání, ale pro snadnější administraci a zobrazování obsahu je potřeba nainstalovat grafické rozhraní jako například `phpmyadmin`. K ulehčení instalace je nezbytné nainstalovat `LAMP server` prostřednictvím `tasksel` balíčku, čehož lze dosáhnout následujícími příkazy

```
sudo apt install tasksel
sudo tasksel install lamp-server
```

Zkratka `LAMP` označuje Linux, Apache2, MySQL a PHP, což jsou nezbytné komponenty pro instalaci balíčku `phpmyadmin`. Samotný balíček potom lze nainstalovat jednoduše

```
sudo apt install phpmyadmin
```

Během instalace je uživatel vyzván k určení webového serveru, na kterém `phpmyadmin` bude fungovat. V tomto případě je zvolen server apache2 a dále je nutné, aby uživatel

zadal heslo k MySQL databázi. Instalace je ukončena a aby bylo možné přistoupit k databázi pomocí `phpmyadmin` rozhraní, je nutné restartovat apache2 příkazem

```
sudo systemctl restart apache2
```

Nyní lze administrovat nebo zobrazovat obsah MySQL databáze přes internetový prohlížeč na adrese `IP_adresa_serveru/phpmyadmin`. Přístup k databázi ovšem funguje na protokolu HTTP, což není šifrovaný protokol a je tedy nezbytná konfigurace serveru `apache2` a `phpmyadmin` za účelem používání šifrovaného protokolu HTTPS podle návodu [37]. Po úspěšné konfiguraci je MySQL databáze připravena se všemi nezbytnými komponenty k plnohodnotnému využívání jako centralizovaná databáze k ukádání logů vytvořených detektorem Raspberry Pi.

Během vypracovávání této bakalářské práce byl vedoucím práce doporučen jiný způsob ukládání a posílání logů do centralizované databáze a to v podobě `Elastic stack`.

2.1.6 Databáze logů – Elastic stack

Elastic stack je skupina open source programů od společnosti Elastic k ukládání, analyzování a vizualizaci dat z jakéhokoliv zdroje, v jakémkoliv formátu. Je nainstalována v operačním systému Ubuntu 20.04, jenž běží ve virtuálním prostředí VirtualBox. Tato skupina obsahuje:

- Elasticsearch
- Logstash
- Kibana
- Beats

Elasticsearch

`Elasticsearch` je vyhledávací a analytický nástroj pro jakýkoliv typ dat včetně textových, numerických nebo lokalizačních, což z něho dělá ústřední součást Elastic stack. Jeho instalace a konfigurace je relativně jednoduchá. Jelikož Elastic podepisuje všechny své balíčky svým podepisovacím klíčem, je nutné nejdříve stáhnout a importovat jejich veřejný klíč k ověření správnosti.

```
wget -qO - www.artifacts.elastic.co/ \
GPG-KEY-elasticsearch | sudo apt-key add -
```

Dále už pouze stačí přidat do systému úložiště, kde se nachází `elasticsearch` balíček a následně ho nainstalovat.

```
echo "deb␣https://artifacts.elastic.co/packages/7.x/␣\
apt␣stable␣main" | sudo tee -a /etc/apt/sources.list.d/ \
elastic-7.x.list
sudo apt update
sudo apt install elasticsearch
```

Jelikož se program po úspěšné instalaci automaticky nespustí, je potřeba ho zapnout manuálně a nastavit, aby se automaticky spouštěl při každém zapnutí počítače.

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo systemctl start elasticsearch.service
```

Po úspěšné instalaci je potřeba program nakonfigurovat, aby se k němu bylo možné připojit i ze vzdáleného počítače. Toto nastavení lze provést změnami v souboru `/etc/elasticsearch/elasticsearch.yml`. Potřebné nastavení lze vidět ve výpise 2.3

Výpis 2.3: Konfigurace programu elasticsearch

```
# ------------------------ Cluster -----------------------
cluster.name: my-application
# ------------------------- Node -------------------------
node.name: node-1
# ----------------------- Network ----------------------
network.host: 0.0.0.0
# ---------------------- Discovery ---------------------
cluster.initial_master_nodes: ["node-1", "node-2"]
```

Po provedení změn je nutné `elasticsearch` restartovat, aby byly změny aplikovány.

```
sudo systemctl stop elasticsearch.service
sudo systemctl start elasticsearch.service
```

Jestli vše proběhlo v pořádku, lze zkontrolovat dotazem na port 9200 zařízení, na kterém `elasticsearch` běží. Pokud je vrácena odpověď podobná jako ve výpise 2.4, znamená to, že `elasticsearch` byl nainstalován a nakonfigurován úspěšně.

Výpis 2.4: Odpověď programu elasticsearch

```
{
  "name" : "node-1",
  "cluster_name" : "my-application",
  "cluster_uuid" : "J1Y_UTPPTa6ihUri0-rx7A",
  "version" : {
    "number" : "7.7.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : 81a1e9eda8e6183f5237786246f6dced26a \
                   10eaf,
    "build_date" : "2020-05-12T02:01:37.602180Z",
    "build_snapshot" : false,
    "lucene_version" : "8.5.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You␣Know,␣for␣Search"
}
```

Logstash

`Logstash` je nenáročný program pro zpracovávání dat z různých zdrojů. Dokáže je transformovat a odesílat do požadovaného cíle. Nejčastěji se používá v kombinaci právě s programem `elasticsearch`. Jelikož lze data posílat přímo do `elasticsearch`, a toto experimentální pracoviště není natolik komplexní, aby vyžadovalo přítomnost programu `logstash`, není jeho instalace nutná.

Kibana

`Kibana` funguje jako grafické rozhraní pro přístup do `elasticsearch` a poskytuje možnosti vyhledávání a vizualizace dat v něm obsažených. Instalace je velice podobná té předchozí. Protože je veřejný Elastic klíč již naimportovaný a úložiště, kde se nacházejí všechny Elastic balíčky přidáno do systému, tyto kroky lze přeskočit a rovnou stáhnout `kibana` balíček.

```
sudo apt install kibana
```

`Kibanu` je také nutné manuálně zapnout a nastavit automatické spouštění při zapnutí počítače.

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable kibana.service
sudo systemctl start kibana.service
```

Jako poslední krok je nezbytná konfigurace souboru `/etc/kibana/kibana.yml` a to konkrétně IP adresy serveru, na kterém běží `elasticsearch`, a povolení připojení ze vzdáleného počítače. Potřebné změny jsou zaznamenány ve výpise 2.5.

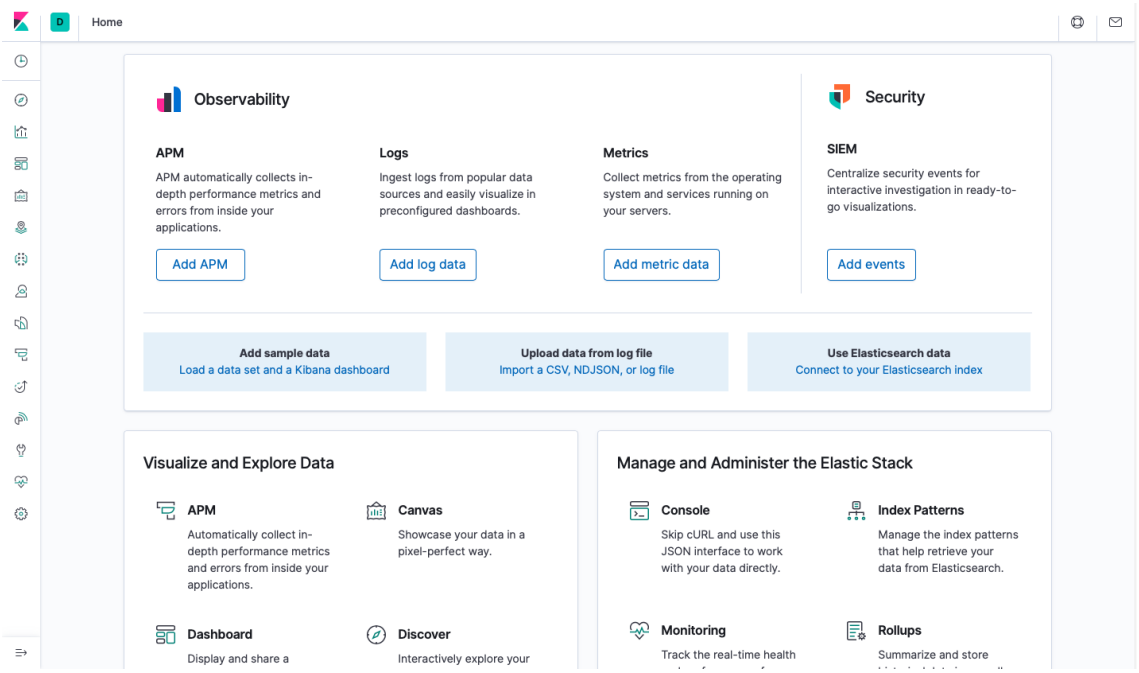
Výpis 2.5: Konfigurace programu kibana

```
server.port: 5601
server.host: "192.168.88.247"
elasticsearch.hosts: ["https://192.168.88.247:9200"]
```

`server.port` je port, na kterém bude `kibana` dostupná, `server.host` je IP adresa serveru, na kterém kibana běží a `elasticsearch.hosts` je seznam IP adres serverů, na kterých běží `elasticsearch`. Po provedení všech potřebných změn v konfiguračním souboru je nezbytné program restartovat.

```
sudo systemctl stop kibana.service
sudo systemctl start kibana.service
```

V tento moment by mělo být vše úspěšně nastaveno a pro přístup do `kibany` stačí zadat IP adresu serveru, na kterém běží, a port 5601. Úvodní stránka je zobrazena na obrázku 2.7



Obr. 2.7: Úvodní stránka kibany

Beats

`Beats` je skupina nenáročných programů určených k posílání dat z různých zařízení do centralizovaného úložiště `logstash` nebo `elasticsearch`. Existuje několik druhů:

- **Filebeat** – Logové a jiné soubory
- **Metricbeat** – Systémové statistiky a statistiky aplikací
- **Packetbeat** – Síťový provoz
- **Winlogbeat** – Logové záznamy Windows
- **Auditbeat** – Uživatelská aktivita a procesy v systémech Linux
- **Heartbeat** – Kontrola, zda-li je systém zapnutý
- **Functionbeat** – Data z cloudu

Pro potřeby tohoto experimentálního pracoviště bude využit `filebeat` pro posílání logových souborů z detektoru do centralizované databáze `elasticsearch`. Raspberry Pi ovšem funguje na architektuře `armhf`, jenž není podporována oficiálním vydavatelem tohoto programu, tudíž se musí stáhnout taková verze, která byla speciálně sestavena pro tuto architekturu. Proto je nejdříve potřeba importovat veřejný klíč vydavatele, který tuto verzi sestavil, a poté přidat do systému úložiště, kde je `filebeat` uložen.

```
wget -O - www.raw.githubusercontent.com/RaoulDuke-Esq \
/Beats-Pi/master/beats-pi.gpg.key | sudo apt-key add -
echo "deb␣https://raw.githubusercontent.com␣\
/RaoulDuke-Esq/Beats-Pi/master␣buster␣main" | sudo tee \
-a /etc/apt/sources.list.d/beats-pi.list
```

Poté již stačí balíček stáhnout jednoduše příkazy

```
sudo apt update
sudo apt install filebeat
```

Před samotným spuštěním je nezbytné `filebeat` nakonfigurovat v souboru `/etc/filebeat/filebeat.yml`. Je potřeba určit, jaké soubory budou posílány do databáze a na jaké IP adrese se databáze nachází. Změny v konfiguračním souboru jsou vypsané ve výpise 2.6.

Výpis 2.6: Konfigurace programu filebeat

```
#==================== Filebeat inputs ==================
filebeat.inputs:
- type: log
  enabled: true
  paths:
     - /cesty/k/logovacim/souborum
#========================= Kibana ======================
setup.kibana:
  host: "192.168.88.247:5601"
#========================= Outputs =====================
output.elasticsearch:
  hosts: ["192.168.88.247:9200"]
```

V sekci `Kibana` je specifikována IP adresa a port, na které je `Kibana` dostupná, a v sekci `Outputs` je vybrán `elasticsearch` a jeho IP adresa a port. `Filebeat` je připraven ke spuštění příkazem

```
sudo systemctl start filebeat
```

Při každém přidání logů do logovacích souborů specifikovaných v konfiguraci je `filebeat` odešle do databáze `elasticsearch`.

2.2 Detekce DoS útoku ICMP flood

Pro otestování, zda-li byla Suricata nakonfigurována a spuštěna správně, byl zvolen DoS útok ICMP flood, jenž byl popsán v kapitole 1.4.2. Útočník generuje obrovské množství ICMP paketů a posílá je na adresu oběti. Signatura byla tedy navržena tak, aby dokázala detekovat frekvenci přicházejících ICMP paketů na adresu oběti, a pokud jich zaznamená více než sto během pěti sekund na stejnou adresu lokalizovanou v domácí síti, začne další pakety zahazovat. Navržená signatura je zobrazena ve výpise 2.7.

Výpis 2.7: Signatura pro detekci útoku ICMP flood

```
drop icmp any any -> $HOME_NET any \
(msg:"Probiha␣ICMP␣flood!"; threshold: type both, \
track by_dst, count 100, seconds 5; itype:8; icode:0; \
classtype:misc-activity; sid:1; rev:1;)
```

Samotný útok se dá realizovat pomocí programu `hping3` a to příkazem

```
hping3 -1 --flood --rand-source <IP adresa oběti>
```

-1 označuje ICMP mód, --flood říká, aby se pakety odesílaly, jak nejrychleji je to možné a --rand-source generuje náhodné zdrojové IP adresy.

Po spuštění útoku na legitimního uživatele, tedy počítač s Windows 10 Pro, začne Suricata běžící na detektoru v IDS módu generovat upozornění, které lze vidět na obrázku 2.8.

May 21, 2020	Message
18:35:38.357	05/21/2020-18:35:38.019637 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 17.208.122.254:8 -> 192.168.88.249:0
18:35:43.018	05/21/2020-18:35:43.004868 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 236.53.5.250:8 -> 192.168.88.249:0
18:35:51.372	05/21/2020-18:35:48.018471 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 153.199.79.175:8 -> 192.168.88.249:0
18:35:58.413	05/21/2020-18:35:53.006061 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 184.72.245.82:8 -> 192.168.88.249:0
18:35:59.103	05/21/2020-18:35:58.031075 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 127.22.196.7:8 -> 192.168.88.249:0
18:36:06.104	05/21/2020-18:36:03.007829 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 142.112.75.19:8 -> 192.168.88.249:0
18:36:09.292	05/21/2020-18:36:08.007379 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 127.82.112.240:8 -> 192.168.88.249:0
18:36:13.746	05/21/2020-18:36:13.006012 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 44.56.183.61:8 -> 192.168.88.249:0
18:36:20.747	05/21/2020-18:36:18.011749 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 232.109.19.221:8 -> 192.168.88.249:0
18:36:24.104	05/21/2020-18:36:23.017493 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 187.32.52.235:8 -> 192.168.88.249:0
18:36:32.818	05/21/2020-18:36:28.017631 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 81.24.134.207:8 -> 192.168.88.249:0
18:36:35.669	05/21/2020-18:36:33.005994 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 0.47.196.5:8 -> 192.168.88.249:0
18:36:38.669	05/21/2020-18:36:38.018315 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 11.67.123.182:8 -> 192.168.88.249:0
18:36:44.185	05/21/2020-18:36:43.008195 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 171.61.39.0:8 -> 192.168.88.249:0
18:36:51.629	05/21/2020-18:36:48.011024 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 122.95.196.222:8 -> 192.168.88.249:0
18:36:56.740	05/21/2020-18:36:53.015948 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 157.11.41.12:8 -> 192.168.88.249:0
18:36:58.267	05/21/2020-18:36:58.005878 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 0.82.28.53:8 -> 192.168.88.249:0
18:37:03.275	05/21/2020-18:37:03.010658 [wDrop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 21.67.23.105:8 -> 192.168.88.249:0

Obr. 2.8: Upozornění vygenerované IDS Suricatou

Pokud je ovšem Suricata spuštěna v IPS módu, pakety nebudou zahazovány, jelikož u funkce port mirroring jsou pakety pouze kopírovány na cílové rozhraní, nikoliv přesměrovány. V okamžiku, kdy se Dos útok zacílí na samotný detektor se Suricatou běžící v IPS módu, pakety budou zahazovány a Suricata bude generovat upozornění viditelné na obrázku 2.9.

May 21, 2020	Message
18:32:48.738	05/21/2020-18:19:35.487925 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 149.189.194.192:8 -> 192.168.88.254:0
18:32:48.738	05/21/2020-18:19:40.007076 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 143.236.82.158:8 -> 192.168.88.254:0
18:32:48.738	05/21/2020-18:19:45.020026 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 32.54.248.236:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:19:50.008401 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 161.78.116.220:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:19:55.011361 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 61.87.129.51:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:00.012308 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 46.85.123.243:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:05.012335 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 142.31.41.124:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:10.012393 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 186.30.237.195:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:15.009436 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 246.78.109.192:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:20.011443 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 189.193.191.66:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:25.013067 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 142.109.161.238:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:30.017814 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 32.235.117.116:8 -> 192.168.88.254:0
18:32:48.739	05/21/2020-18:20:35.015149 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 165.222.209.111:8 -> 192.168.88.254:0
18:32:48.740	05/21/2020-18:20:40.010275 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 243.121.90.10:8 -> 192.168.88.254:0
18:32:48.740	05/21/2020-18:20:45.009659 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 166.171.233.213:8 -> 192.168.88.254:0
18:32:48.740	05/21/2020-18:20:50.014945 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 85.24.124.193:8 -> 192.168.88.254:0
18:32:48.740	05/21/2020-18:20:55.006862 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 12.17.165.183:8 -> 192.168.88.254:0
18:32:48.740	05/21/2020-18:21:00.014199 [Drop] [**] [1:1:1] Probiha ICMP flood! [**] [Classification: Misc activity] [Priority: 3] {ICMP} 51.114.165.94:8 -> 192.168.88.254:0

Obr. 2.9: Upozornění vygenerované IPS Suricatou

Suricata detekovala útok podle předpokladů, tudíž lze začít s detekcí útoku MitM.

2.3 Detekce MitM útoku ARP spoofing

Protokol ARP funguje na spojové vrstvě modelu ISO/OSI, ale Suricata dokáže analyzovat pakety pouze od síťové vrstvy nahoru. Ani po vyčerpávajícím testování konfiguračních souborů a rozsáhlém pátrání po různých zdrojích se nepodařilo nastavit Suricatu, aby dokázala pracovat nad spojovou vrstvou, tudíž ji lze označit jako nevhodnou pro detekci útoku ARP spoofing. Byl tedy navržen skript v programovacím jazyku Python, běžící na detektoru Raspberry Pi, který je schopný kontrolovat pakety i na druhé vrstvě modelu ISO/OSI. Využívá několik naimportovaných modulů a to konkrétně:

- **Scapy** – nástroj pro manipulaci paketů, může například odesílat, falšovat nebo zachytávat pakety v síti. Ve skriptu je využit k zachytávání a následné filtraci ARP paketů.
- **Subprocess** – modul, jenž je používán ke spouštění nových aplikací nebo programů vytvářením nových procesů. Je využit k navazování spojení se směrovačem Mikrotik prostřednictvím šifrovaného protokolu SSH (Secure SHell), což je nezbytné k vykonávání příkazů a skriptů na směrovači.
- **Logging** – standardní pythonovský modul určený k vytváření logů. Ve skriptu je využit pouze v případě, kdy není dostupná MySQL databáze, tudíž se logovací záznamy ukládají do lokálního souboru na detektoru.
- **Os** – standardní modul k interakci s operačním systémem, na kterém pythonovský skript aktuálně běží. Je využit pro kontrolu, zda byl skript spuštěn s právy uživatele root, což je nezbytné pro práci s internetovými rozhraními na detektoru.

Skript komunikuje se směrovačem Mikrotik prostřednictvím šifrovaného protokolu SSH. Ten ovšem požaduje zadání hesla pokaždé, když je toto spojení navázano, což není žádoucí, poněvadž by byl uživatel spouštějící skript nucen opakovaně zadávat přístupové heslo do směrovače, a proto je nutné na detektoru vytvořit dvojici RSA klíčů, soukromý a veřejný klíč. To je možné provét příkazem

```
ssh-keygen -t rsa
```

Po úspěšném vygenerování klíčů se veřejný klíč musí nahrát na směrovač například pomocí protokolu FTP (File Transfer Protocol) a následně naimportovat, aby to nebyl pouze soubor uložený na Mikrotiku, nýbrž platný veřejný klíč, jenž se dá použít při autentizaci uživatele v rámci protokolu SSH. Na Mikrotiku je proto nutné zadat tento příkaz

```
/user ssh-keys import public-key-file=id_rsa.pub
```

, kde public-key-file určuje název souboru s veřejným klíčem.

Pokud je veřejný klíč úspěšně naimportován na směrovač, je vše připraveno a skript je plně funkční. Jako první akci, kterou skript vykoná po spuštění je ověření, zda byl spuštěn s právy uživatele root, což je nutné pro práci s internetovými rozhraními. Pokud vyhodnotí, že jsou práva nedostačující, ukončí se. V opačném případě nabídne uživateli seznam dostupných internetových rozhraní a vyzve ho k vybrání jednoho, na kterém bude zachytávat pakety. Poté je již inicializována funkce `sniff` knihovny `Scapy`, jenž lze vidět ve výpisu 2.8, která se stará o zachytávání paketů.

Parametry této funkce jsou `iface`, což je dříve zvolené rozhraní uživatelem, na kterém budou pakety zachycovány, parametr `filter` určuje, jaké pakety budou zachycovány. Hodnota `arp` udává, že budou zachycovány pouze ARP pakety. Parametr `prn` určuje funkci, která je zavolána v případě, kdy je zachycen paket vyhovující všem filtrům a poslední parametr `store` udává, zda-li budou zachycené pakety uloženy v paměti. Toto nastavení je užitečné v případě, kdy je skript spuštěn dlouhou dobu.

Výpis 2.8: Inicializace funkce sniff

```
scapy.sniff(iface=iface, filter="arp", prn=packet_filter,
            store=0)
```

Jak již bylo zmíněno dříve, funkce `sniff` volá metodu `packet_filter`. Ta se stará o podrobnější filtraci paketů a extrahuje potřebné informace přímo z paketu. Konkrétně přijímá pouze ARP reply pakety a z paketu získává zdrojovou a cílovou MAC adresu. V neposlední řadě volá metodu `get_mac`, která má za úkol poslat ARP request paket na IP adresu, ze které přišel zkoumaný paket ARP reply. Po obdržení odpovědi vrací metodě `packet_filter` zodpovězenou zdrojovou MAC adresu. Ta tedy aktuálně obsahuje zdrojovou a cílovou MAC adresu paketu, která je ve skriptu označená jako `falesna_mac` a `cilova_mac`, a zodpovězenou MAC adresu zařízení, ze kterého daný paket přišel, takzvaná `realna_mac`. Všechny tyto informace předává nejkomplexnější metodě zvané `detekce`, kde se uskutečňuje rozhodování, zda-li je daný paket škodlivý či nikoliv. Rozhoduje se dvěma různými způsoby. Nejdříve zkontroluje, zda-li ARP tabulka na směrovači Mikrotik neobsahuje duplicitní hodnotu zdrojové MAC adresy paketu. Jestliže duplicitní hodnotu nalezne, znamená to, že v síti probíhá dvoustranný ARP spoofing útok na některé zařízení, což znamená, že útočník posílá škodlivé ARP reply pakety nejen oběti, ale také směrovači. Ke kontrole duplicitního záznamu je nutná komunikace se směrovačem. To zajišťuje protokol SSH, díky němuž je na Mikrotiku spuštěn skript, jenž lze vidět ve výpise 2.9, který vytvoří proměnou, obsahující všechny ARP záznamy s MAC adresou `falesna_mac` a vrátí délku této proměnné, nebo-li počet ARP záznamů uložených v dané proměnné. Pokud obsahuje dva a více záznamů, lze tvrdit, že na síti probíhá útok.

Výpis 2.9: Mikrotik skript k zjištění duplicitních záznamů v ARP tabulce

```
:global array [:toarray ""];
:foreach i in=[/ip arp find where \
mac-address=<falesna_mac>] \
do=[:set array ($array, $i)]
;:return [:len $array]
```

Pokud ovšem duplicitní záznam nalezen není, metoda zkusí porovnat zdrojovou MAC adresu paketu označenou jako `falesna_mac` se zdrojovou adresou zodpovězenou zařízením, které paket odeslalo, označenou jako `realna_mac`. Pokud se tyto dvě adresy nebudou shodovat, znamená to, že na síti probíhá jednostranný ARP spoofing útok, jelikož paket odeslalo jiné zařízení, než bylo uvedeno v hlavičce paketu.

V momentě, kdy metoda vyhodnotí, že útok ARP spoofing probíhá, počká na desátý škodlivý paket, aby se předešlo blokaci legitimních zařízení a je zavolána metoda `blok`. Tato metoda se stará o odeslání instrukcí směrovači k zablokování komunikace nežádoucí MAC adresy. Jsou odeslány čtyři instrukce:

1. Vytvořit nový subnet 192.168.66.0/24 v záložce IP/Addresses
2. Vytvořit nový subnet 192.168.66.0/24 v záložce IP/DHCP Server/Networks
3. Vytvořit nové pravidlo ve firewallu, které zahazuje veškerou komunikaci přicházející od nežádané MAC adresy v záložce IP/Firewall/Filter Rules
4. Přiřadit nežádoucí MAC adresu do nově vytvořeného subnetu 192.168.66.0/24 v záložce IP/DHCP Server/Leases

Přesně zadané příkazy jsou viditelné ve výpise 2.10.

Výpis 2.10: Blokace komunikace nežádoucí MAC adresy na směrovači Mikrotik

```
/ip address add address=192.168.66.1/24 \
network=192.168.66.0 interface=bridge;
/ip dhcp-server network add address=192.168.66.0/24 \
gateway=192.168.66.1;
/ip firewall filter add chain=forward \
src-mac-address=<falesna_mac> action=drop place-before=1;
:foreach i in=[/ip dhcp-server lease find where \
mac-address=<falesna_mac>] do={[/ip dhcp-server lease \
make-static $i];[/ip dhcp-server lease set $i \
address=192.168.66.66]}
```

Pouhé vytvoření pravidla zahazující provoz přicházející z konkrétní MAC adresy by bylo nedostačující. Mělo by to pouze za následek omezení komunikace daného zařízení se zařízeními nacházejících se v jiných sítích. Ovšem komunikace se zařízeními ve stejné síti by byla stále možná, protože v okamžiku, kdy jsou dvě zařízení ve

stejném subnetu, směrovač provoz mezi těmito zařízeními nesměruje, nýbrž pouze přepíná, takže pakety nejsou porovnávány s pravidly ve firewallu. Z tohoto důvodu je nezbytné kromě vytvoření pravidla ve firewallu, také vytvoření nového subnetu a přiřazení do něj nežádoucí MAC adresy. Tím se zamezí komunikaci s jakýmkoliv jiným zařízením.

Skript byl napsán v programovacím jazyku python 3, proto je důležité ho spouštět v příslušné verzi. Spuštění skriptu je velice jednoduché a lze provést jediným příkazem.

```
sudo python3 /cesta/ke/skriptu/arpdetector.py
```

Po spuštění je nutné vybrat rozhraní, na kterém budou detekovány ARP spoofing útoky a poté je skript funkční.

```
Zadejte detekovane rozhrani. ['lo', 'wlan0', 'eth0']
eth0
ARP poisoning detekce zacala na adrese <IP adresa>
```

Pro testování skriptu byl zvolen útok ARP spoofing popsaný v kapitole 1.4.1. Ten se dá na útočníkově počítači Kali Linux vykonat pomocí programu `bettercap`. Spuštění programu lze provést následujícím příkazem.

```
root@kali:~# bettercap -iface eth0
```

Parametr `iface` určuje internetové rozhraní. Po spuštění je nutné nastavit několik proměnných. Konkrétně se jedná o `arp.spoof.fullduplex`, jenž se nastavuje pouze tehdy, pokud je požadovaný dvoustranný MitM útok, kdy se neútočí pouze na oběť, ale zároveň i na směrovač v síti. Nastavení této proměnné by bylo ignorováno v případě, kdy by byl žádaný pouze jednostranný útok. Dále je potřeba nastavit cíle, na které bude útok směrován. To lze uskutečnit nastavením proměnné `arp.spoof.targets`. Poté lze již iniciovat útok.

```
set arp.spoof.fullduplex true
set arp.spoof.targets <IP adresa oběti>
arp.spoof on
```

Vygenerované logy během dvoustranného ARP spoofing útoku jsou viditelné na obrázku 2.10 a jednostranného útoku na obrázku 2.11.

Z logových záznamů lze vyčíst, že skript funguje podle předpokladů. Úspěšně detekoval MitM útok mířený na počítač oběti a při detekci desátého škodlivého paketu úspěšně tento útok mitigoval.

May 21, 2020	Message
17:46:49.034	INFO: 21.05.2020 17:46:45: ARP poisoning detekce zacala na adrese 192.168.88.254
17:47:45.304	WARNING: 21.05.2020 17:47:44: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 0
17:47:46.585	WARNING: 21.05.2020 17:47:45: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 74:4D:28:BB:29:FB. Upozorneni cislo 1
17:47:49.586	WARNING: 21.05.2020 17:47:47: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 2
17:47:50.881	WARNING: 21.05.2020 17:47:48: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 74:4D:28:BB:29:FB. Upozorneni cislo 3
17:47:50.882	WARNING: 21.05.2020 17:47:50: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 4
17:47:51.883	WARNING: 21.05.2020 17:47:51: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 74:4D:28:BB:29:FB. Upozorneni cislo 5
17:47:52.883	WARNING: 21.05.2020 17:47:52: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 6
17:47:54.181	WARNING: 21.05.2020 17:47:53: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 74:4D:28:BB:29:FB. Upozorneni cislo 7
17:47:54.865	WARNING: 21.05.2020 17:47:54: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu DC:A6:32:16:F3:30. Upozorneni cislo 8
17:47:57.865	WARNING: 21.05.2020 17:47:56: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu DC:A6:32:16:F3:30. Upozorneni cislo 9
17:47:58.039	WARNING: 21.05.2020 17:47:57: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 10
17:47:58.039	INFO: 21.05.2020 17:47:57: MAC adrese 08:00:27:B4:94:03 byla zablokovana komunikace s ostatnimi klienty v siti.
17:48:01.040	WARNING: 21.05.2020 17:47:59: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 74:4D:28:BB:29:FB. Upozorneni cislo 11
17:48:01.197	WARNING: 21.05.2020 17:47:59: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu DC:A6:32:16:F3:30. Upozorneni cislo 12
17:48:01.197	WARNING: 21.05.2020 17:48:00: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 13
17:48:02.198	WARNING: 21.05.2020 17:48:01: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 74:4D:28:BB:29:FB. Upozorneni cislo 14
17:48:03.842	WARNING: 21.05.2020 17:48:03: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 15

Obr. 2.10: Upozornění vygenerované Python skriptem během dvoustranného ARP spoofing útoku

May 21, 2020	Message
17:56:48.445	INFO: 21.05.2020 17:56:38: ARP poisoning detekce zacala na adrese 192.168.88.254
17:57:53.639	WARNING: 21.05.2020 17:57:52: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 0
17:57:54.188	WARNING: 21.05.2020 17:57:54: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 1
17:57:57.189	WARNING: 21.05.2020 17:57:56: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 2
17:57:59.122	WARNING: 21.05.2020 17:57:58: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 3
17:58:13.365	WARNING: 21.05.2020 17:58:01: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 4
17:58:13.365	WARNING: 21.05.2020 17:58:02: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 5
17:58:13.366	WARNING: 21.05.2020 17:58:04: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 6
17:58:13.367	WARNING: 21.05.2020 17:58:07: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 7
17:58:13.367	WARNING: 21.05.2020 17:58:09: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 8
17:58:13.367	WARNING: 21.05.2020 17:58:11: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 9
17:58:16.368	WARNING: 21.05.2020 17:58:14: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 10
17:58:22.634	INFO: 21.05.2020 17:58:14: MAC adrese 08:00:27:B4:94:03 byla zablokovana komunikace s ostatnimi klienty v siti.
17:58:22.634	WARNING: 21.05.2020 17:58:15: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 11
17:58:22.635	WARNING: 21.05.2020 17:58:17: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 12
17:58:22.635	WARNING: 21.05.2020 17:58:20: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 13
17:58:29.636	WARNING: 21.05.2020 17:58:26: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 14
17:58:29.892	WARNING: 21.05.2020 17:58:27: ARP poisoning detekovano na MAC adrese 08:00:27:B4:94:03 mireno na adresu 60:EB:69:CF:10:1D. Upozorneni cislo 15

Obr. 2.11: Upozornění vygenerované Python skriptem během jednostranného ARP spoofing útoku

2.4 Arpwatch

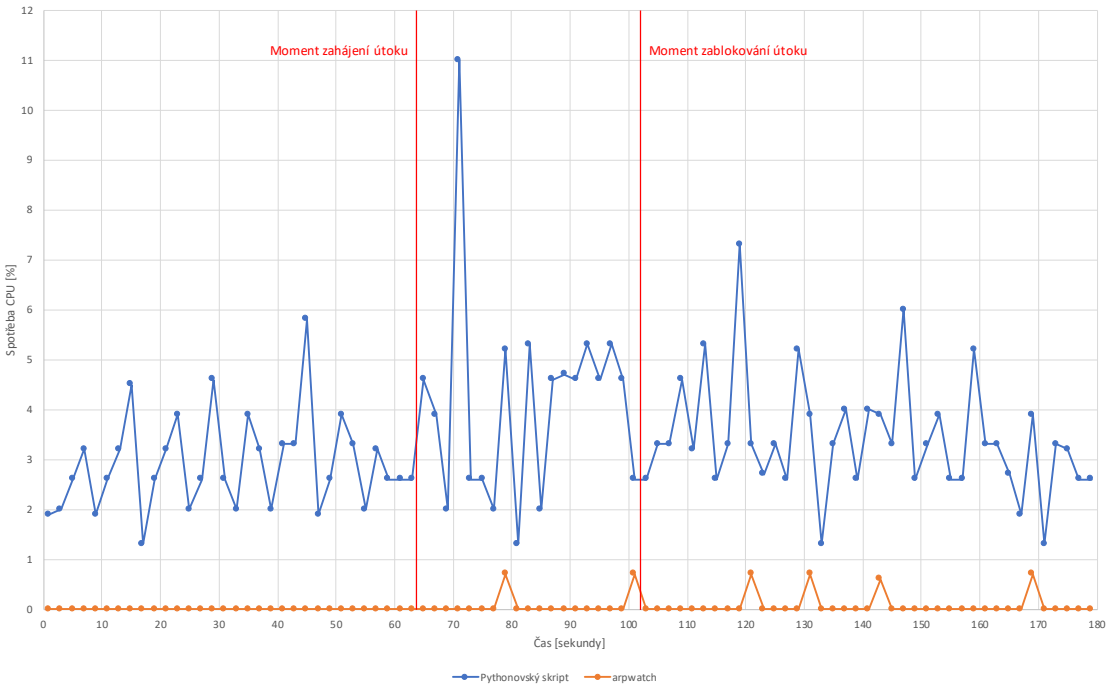
Velice užitečný nástroj určený k monitorování lokální sítě je `arpwatch`. Udržuje si databázi dvojic IP a MAC adres, takže dokáže detekovat, pokud si nějaké zařízení mění buď IP nebo MAC adresu. To je velmi nápomocné, jelikož přesně takové chování je příznakem ARP spoofing útoku. Jestliže je podobné chování detekováno nebo pokud se v síti pouze objevilo nové zařízení, lze `arpwatch` nastavit, aby odeslal report na předem definovanou emailovou adresu. Jeho instalace je velice snadná, jelikož se nachází v balíčkové databázi. Stačí zadat příkaz

```
sudo apt install arpwatch
```

Před prvním spuštěním `arpwatch` je nutné vytvořit soubor `/var/lib/arpwatch/arp.dat`, kam si bude program ukládat dvojice IP a MAC adres. Po vytvoření tohoto souboru je možné program spustit příkazem

```
sudo arpwatch -i <monitorované rozhraní>
```

Tento program ovšem nedisponuje funkcí mitigace takovýchto útoků, a proto nebyl zvolen k využití v této bakalářské práci. Byly ovšem otestovány jeho požadavky na výpočetní výkon detektoru a porovnány s požadavky pythonovského skriptu viditelné na obrázku 2.12. Oba dva programy byly testovány souběžně. Nejdříve byly spuštěny bez přítomnosti ARP spoofing útoku a přibližně v polovině měření byl útok spuštěn. Ten byl následně zablokován pythonovským skriptem a měření pokračovalo. Momenty spuštění a zablokování útoku jsou znázorněny v již zmíněném grafu.



Obr. 2.12: Porovnání procentuální spotřeby CPU pythonovského skriptu a arpwatch

Z grafu lze vyčíst, že program `arpwatch` je velice nenáročný program co se týče výpočetního výkonu. Během celého měření využil průměrně 0,0456 % výpočetního výkonu. Pythonovský skript na druhou stranu využíval průměrně 3,3878 % a v okamžiku spuštění kybernetického útoku dokonce 11 %. Procentuální spotřeba paměti RAM nebyla v grafu znázorněna, jelikož byla v průběhu celého měření konstantní. Arpwatch spotřebovával 0,1 % paměti RAM a pythonovský skript 1,5 %.

Závěr

Cílem této bakalářské práce bylo analyzování současného stavu problematiky, realizování experimentálního pracoviště a návrh a implementace detekce útoku Man-in-the-Middle.

Analyzování současného stavu problematiky se věnovala teoretická část semestrální práce, kde byla detailně popsána analýza sítě a nástroje k tomu využívané. Tyto nástroje byly potom rozděleny podle jejich umístění nebo způsobu analýzy. Dále byly rozebrány kybernetické útoky a to konkrétně útoky Man-in-the-Middle a Denial-of-Service.

Realizování experimentálního pracoviště byl rozebráno v praktické části, kde byly detailně popsány jednotlivé komponenty včetně jejich technických parametrů. Tato část se také věnovala instalaci a konfiguraci databáze MySQL určené k ukládání logových záznamů, jenž byla během vypracovávání této práce nahrazena databází Elasticsearch, jenž byla také popsána. Tato část se též věnovala programu Suricata, sloužící k analýze síťového provozu a detekci útoků Denial-of-Service a Man-in-the-Middle.

Pro otestování, zda-li byla Suricata nakonfigurována a spuštěna správně, byl zvolen DoS útok ICMP flood. Byla tedy navržena signatura na detekci daného útoku. V momentě, kdy Suricata běžela v IDS módu, byla schopna detekovat tento útok, ať už byl cílený na legitimního uživatele nebo přímo na detektor. Pokud byla ale Suricata zapnuta v IPS módu, dokázala tento útok mitigovat pouze, pokud byl cílený přímo na detektor, nikoliv na legitimního uživatele. Aby detektor mohl mitigovat útoky, které nebyly cíleny přímo na něj, bylo by potřeba, aby měl minimálně dvě síťová rozhraní. Pokud by tomu tak bylo, příchozí provoz by přicházel do jednoho rozhraní, Suricata by následovně provoz analyzovala a vše, co by bylo schváleno jako legitimní provoz, by odcházelo druhým rozhraním. V případě experimentálního pracoviště použitého v této bakalářské práci to nebylo možné, jelikož jako detektor bylo použito Raspberry Pi, jenž má pouze jedno síťové rozhraní. Vše ovšem fungovalo podle předpokladů, a proto bylo možné začít testování mitigace MitM útoku ARP spoofing. Vyskytl se ovšem problém, že tento útok funguje na spojové vrstvě modelu ISO/OSI a Suricata dokáže pracovat pouze nad vrstvami 3 až 7. Byla tudíž navržena vlastní implementace detekce v programovacím jazyku Python.

Tato implementace již dokázala pracovat s pakety na spojové vrstvě díky nástroji Scapy, tudíž bylo možné úspěšně útok detekovat. Během detekce byla při každém zachyceném ARP reply paketu kontrolována ARP tabulka na směrovači Mikrotik, zda-li neobsahovala duplicitní záznam MAC adres. Pokud tomu tak nebylo, byl následně odeslán ARP request paket na IP adresu, ze které paket přišel, a pokud se zodpovězená MAC adresa neshodovala se zdrojovou MAC adresu v hlavičce paketu,

byl detekován útok. Následnou mitigaci zajišťoval směrovač vytvořením nového subnetu, do kterého přiřadil MAC adresu, ze které byl útok detekován, a vytvořením nového pravidla ve firewallu zahazující všechny pakety přicházející z dané adresy. Tím pádem se zamezilo komunikaci se všemi zařízeními připojených ke směrovači, jelikož se nacházeli v jiné síti než útočník.

Všechny zadané cíle byly tímto splněny.

Literatura

[1] *What Is a LAN?* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.cisco.com/c/en/us/products/switches/what-is-a-lan-local-area-network.html>

[2] *What is Man-In-The-Middle Attack (MITM) – Prevention* [online]. April 12, 2017 [cit. 2019-12-15]. Dostupné z: <https://www.thewindowsclub.com/man-in-the-middle-attack>

[3] *Vysvětlení síťového modelu ISO/OSI a popis TCP/IP* [online]. 8. 8. 2011 [cit. 2019-12-15]. Dostupné z: <http://hacking.blog.zive.cz/2011/08/vysvetleni-sitoveho-modelu-isoosi-a-popis-tcpip>

[4] *Referenční model ISO/OSI - sedm vrstev* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.earchiv.cz/a92/a213c110.php3>

[5] FAJKUS, Karel. *Detekce anomálií v lokální síti* . Brno, 2018. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Vlastislav Dohnal.

[6] ALAIDAROS, Hashem Mohammed, Massudi MAHMUDDIN a Ali AL-MAZARI. *An Overview of Flow-Based and Packet-Based Intrusion Detection Performance in High Speed Networks* [online]. 2011 [cit. 2019-12-15]. Dostupné z: <https://www.semanticscholar.org/paper/An-Overview-of-Flow-Based-and-Packet-Based-in-High-Alaidaros-Mahmuddin/6b3f09231c8d5391528071cf3fceba3e79c9f7bc#paper-header>

[7] COLLINS, Michael. *Network security through data analysis: building situational awareness*. O'Reilly, 2014. ISBN 978-1-449-35790-0.

[8] MARTINÁSEK, Zdeněk. *8. Přednáška: Systémy IDS a IPS*. Brno: VUT v Brně, 2018.

[9] LA ROSA, Alexander. *Log Monitoring: not the ugly sister* [online]. February 8, 2018 [cit. 2019-12-15]. Dostupné z: <https://pandorafms.com/blog/log-monitoring/>

[10] MAURO, Andrea. *How to Discover Your Network Topology* [online]. 10/10/19 [cit. 2020-05-29]. Dostupné z: <https://blogs.arubanetworks.com/solutions/how-to-discover-your-network-topology/>

[11] *Top 7 Network Mapping Tools in 2020* [online]. September 20, 2019 [cit. 2020-05-29]. Dostupné z: <https://www.dnsstuff.com/network-mapping-tools/>

[12] *What Are the Most Common Cyber Attacks?* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html>

[13] *Learn about cyber attacks and how to defend against them* [online]. 2 October 2018 [cit. 2019-12-15]. Dostupné z: <https://www.ibm.com/services/business-continuity/cyber-attack>

[14] *What is Man-in-the-Middle Attack?* [online]. [cit. 2019-12-15]. Dostupné z: <https://securebox.comodo.com/ssl-sniffing/man-in-the-middle-attack/>

[15] PUBLICO, Ricky. *What is a Man-in-the-Middle Attack and How Can You Prevent It?* [online]. 01 Mar 2017 [cit. 2019-12-15]. Dostupné z: <https://www.globalsign.com/en/blog/what-is-a-man-in-the-middle-attack/>

[16] *Man-in-the-Middle (MITM) Attacks: MITM Techniques, Detection, and Best Practices for Prevention* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/>

[17] *Man in the middle (MITM) attack* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/>

[18] *Man in the middle útok v C# – ARP poisoning (1)* [online]. 2.12.2013 [cit. 2019-12-15]. Dostupné z: <https://www.soom.cz/clanky/1128--Man-in-the-middle-utok-v-C-ARP-poisoning-1>

[19] *What is IP Spoofing?* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/>

[20] TZUR-DAVID, Shimrit. *The Ultimate Guide to Man in the Middle (MITM) Attacks and How to Prevent them* [online]. October 8th, 2018 [cit. 2019-12-15]. Dostupné z: <https://doubleoctopus.com/blog/the-ultimate-guide-to-man-in-the-middle-mitm-attacks-and-how-to-prevent-them

[21] *IP Spoofing: An Introduction* [online]. 10 Mar 2003 [cit. 2019-12-15]. Dostupné z: <https://www.symantec.com/connect/articles/ip-spoofing-introduction>

[22] *What is DNS Cache Poisoning or Spoofing?* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.kaspersky.com/resource-center/definitions/dns>

[23] *DNS SPOOFING (DNS CACHE POISONING)* [online]. [cit. 2019-12-15]. Dostupné z: <https://doubleoctopus.com/security-wiki/threats-and-tools/dns-spoofing/>

[24] *What is DNS cache poisoning? | DNS spoofing* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.cloudflare.com/learning/dns/dns-cache-poisoning/>

[25] *SSL Spoofing: Man-In-The-Middle attack on SSL* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.owasp.org/images/7/7a/SSL_Spoofing.pdf>

[26] ARAMPATZIS, Anastasios. *What Are SSL Stripping Attacks?* [online]. September 10, 2018 [cit. 2019-12-15]. Dostupné z: <https://www.venafi.com/blog/what-are-ssl-stripping-attacks>

[27] *What is SSL Stripping? A Beginner's Guide to SSL Strip Attacks* [online]. [cit. 2019-12-15]. Dostupné z: <https://comodosslstore.com/blog/what-is-ssl-stripping-beginners-guide-to-ssl-strip-attacks.html>

[28] *What is a denial of service attack (DoS) ?* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos>

[29] *Buffer Overflow Attack with Example* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.geeksforgeeks.org/buffer-overflow-attack-with-example/>

[30] *Ping (ICMP) Flood DDoS Attack* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.cloudflare.com/learning/ddos/ping-icmp-flood-ddos-attack/>

[31] *TCP 3-Way Handshake (SYN, SYN-ACK,ACK)* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.guru99.com/tcp-3-way-handshake.html>

[32] STRAKOŠ, Jan. *Detekce anomálií pomocí neuronových sítí*. Brno, 2019, 53 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Petr Blažek

[33] *SYN Flood Attack* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/>

[34] *About* [online]. [cit. 2019-12-15]. Dostupné z: <https://www.mikrotik.com/aboutus>

[35] *What is a Raspberry Pi?* [online]. [cit. 2019-12-15]. Dostupné z: <https://opensource.com/resources/raspberry-pi>

[36] *What is Kali Linux?* [online]. 2019-Oct-26 [cit. 2019-12-15]. Dostupné z: <https://www.kali.org/docs/introduction/what-is-kali-linux/>

[37] CÁNEPA, Gabriel *How to Setup HTTPS (SSL Certificates) to Secure PhpMyAdmin Login* [online]. October 4, 2016 [cit. 2020-05-16]. Dostupné z: <https://www.tecmint.com/setup-https-ssl-certificates-to-secure-phpmyadmin-login/>

[38] *Download* [online]. [cit. 2019-12-15]. Dostupné z: <https://suricata-ids.org/download/>

[39] *Debian Installation* [online]. [cit. 2019-12-15]. Dostupné z: <https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Debian_Installation>

[40] LAIARTURS *RouterOS API* [online]. 23 Aug 2019 [cit. 2020-05-18]. Dostupné z: <https://github.com/LaiArturs/RouterOS_API>

Seznam symbolů, veličin a zkratek

LAN Local Area Network
WAN Wide Area Network
MitM Man-in-the-Middle
ISO/OSI International Organization for Standardization/Open System Interconnection
TPDU Transaction Protocol Data Unit
SPDU Session Protocol Data Unit
PPDU Presentation Protocol Data Unit
APDU Application Protocol Data Unit
IDS Intrusion Detection System
IPS Intrusion Prevention System
DPI Deep Packet Inspection
IP Internet Protocol
SQL Structured Query Language
ARP Address Resolution Protocol
MAC Media Access Control
DNS Domain Name System
HTTPS Hypertext Transfer Protocol Secure
HTTP Hypertext Transfer Protocol
SSL Secure Socket Layer
TCP Transmission Control Protocol
DoS Denial-of-Service
ICMP Internet Control Message Protocol
SYN SYNchronize
ACK ACKnowledgment
DDoS Distributed-Denial-of-Service
RAM Random Access Memmory
VNC Virtual Network Computing
UDP User Datagram Protocol
OISF Open Inforamtion Security Foundation

Seznam příloh

A Obsah přiloženého CD **54**

A Obsah přiloženého CD

- `/` ..... kořenový adresář přiloženého CD
  - `arpdetector.py` ..... skript detekčního sytému
  - `bakalarska-prace-sipek.pdf` ..... elektronická verze práce ve formátu PDF
  - `configs` ..... adresář obsahující konfigurační soubory
    - `elasticsearch.yml` ..... konfigurační soubor programu elasticsearch
    - `filebeat.yml` ..... konfigurační soubor programu filebeat
    - `kibana.yml` ..... konfigurační soubor programu kibana
    - `mikrotik-config.backup` ..... záloha konfigurace směrovače Mikrotik
    - `suricata.yaml` ..... konfigurační soubor programu suricata
  - `images` ..... adresář obsahující obrázky použité v této práci